

OCTUBRE 1987
LOAD MSX
AÑO 2 Nº 18
A5 REP.ARGENTINA
BASIC PARA MSX Y PC
MANEJO DE PANTALLAS
COMO ACELERAR UN PROGRAMA
LENGUAJE C
CONCURSOS Y SORTEOS

EN INFORMATICA TALENT MSX HACE ESCUELA.

Y una prueba de ello, son algunos de los establecimientos que han incorporado computadoras Talent MSX como herramienta de apoyo pedagógico.

CAPITAL FEDERAL:
SIDERCA SAIC
ASOC. CRISTIANA DE JOVENES
COLEGIO JESUS MARIA
ESC. Nº 2 D.F. SARMIENTO
UNIVERSIDAD DEL SALVADOR
C.O.D.I.C.E.
CENEA
C.L.A.I.C.E.
INST. INMACULADA CONCEPCION
FUND. NTRA. SRA. DE LA MERCED
FUND. HNOS. A. Y E. ROCCA
INST. TECNICO DE BS. AS.
ESCUELA ARG. MODELO
COLEGIO ESTEBAN ECHEVERRIA
INST. JOSE MANUEL ESTRADA
ASOCIACION ISRAELITA ARGENTINA
INST. LA INMACULADA
ESC. Nº 24
CTRO. DE INF. PSICOPEDAGOGICA
NTRA. SRA. DE LA MISERICORDIA
ESC. REP. ORIENTAL DEL URUGUAY
ESC. Nº 10
ESC. MODELO D.F. SARMIENTO
INST. NTRA. SRA. DE LOS REMEDIOS
INST. PRIV. SAN CAYETANO
COLEGIO SAN GREGORIO
COL. MARIE MANOOGIAN
ESCUELA Nº 11
ESC. Nº 14 FRANCISCO BEIRO
INST. SAN VICENTE DE PAUL
ESC. Nº11 POR LA NIÑEZ
INSTITUTO BAYARD
LAB. DE COMP. CLINICA Y EDUC.
ESC. Nº 5 URSULA DE LAPUENTE
COLEGIO ISLAS MALVINAS
COL. CHARLES DE FOUCALD
C.O.E.S.O. LTDA.
NTRA. SRA. DEL SAGRADO CORAZON
ESCUELA ARGENTINA 2000
ESC. TEC. RAGGIO
BS. AS. ENGLISH HIGH SCHOOL
ESC. M. N. VIOLA
INST. SAN PIO X
ESCUELA Nº 5
INST. MARIA ANA MOGAS
INST. SUDAMERICANO MODELO
COLEGIO NTRA. SRA. DEL MILAGRO
CESCOM
COL. SGDO. CORAZON
COLEGIO SANTA BRIGIDA
INST. SAN PATRICIO
COLEGIO ANDERSEN
E.N.E.T. Nº 13
PROG. CULT. EN SINDICATO
CIR SUBOF. DE LA POLICIA FED.
NEW MODEL SCHOOL

BUENOS AIRES:
INST. SAN GABRIEL - ADROGUE
ESC. ENS. MEDIA Nº 4 - ALGARROBO
ESCUELA Nº 28 - AVELLANEDA
INST. FRENCH - AVELLANEDA
E.N.E.T. Nº 1 V. PEREDA - AZUL
ESC. ENS. MEDIA Nº 4 - BAHIA BLANCA
COLEGIO DON BOSCO - BAHIA BLANCA
ESC. SUP. DE COMERCIO - BAHIA BLANCA
INST. SUP. JUAN XXIII - BAHIA BLANCA
CTRO. DE EDU. AVANZADA - B. BLANCA
SEARCH S.A. - BAHIA BLANCA
ESCUELA Nº 12 - BERAZATEGUI
JBS INFORMATICA - BERAZATEGUI
ESCUELA Nº 3 - BERISSO
COMPUSISTEM - BERISSO
STA. MARIA DE LAS LOMAS - BOULOGNE
COLEGIO GOODSPELL - BOULOGNE
E.N.E.T. Nº5 C. SARMIENTO - CAPITAN SARMIENTO
ESC. Nº 9 NTRA. S. DEL CARMEN - CARLOS CASARES
ESC. Nº 7 D.F. SARMIENTO - C. CASARES
E.N.E.T. Nº 1 - CARLOS CASARES
ESC. Nº 1 J. M. ESTRADA - C. CASARES
ESC. ENS. MEDIA Nº 1 - CHASCOMUS
CENTRO INF. ESC. Nº 5 - CHASCOMUS
COL. CORAZON DE MARIA - CHASCOMUS
COL. JUAN GALO DE LAVALLE - CHASCOMUS
ESCUELA Nº 1 D.F. SARMIENTO - CORONEL PRINGLES
COLEGIO CRISTO REY - DOCK SUD
ESCUELA Nº28 - DON TORCUATO
ESC. ENS. MEDIA Nº 5 - DON TORCUATO
ESCUELA Nº 14 - ESCOBAR
COLEGIO JESUS MARIA - FCIO. VARELA
INST. LA SALLE - FLORIDA
ACADEMIA COMERCIAL BELGRANO - GRAL. RODRIGUEZ
INST. GRAL. PACHECO - GRAL. PACHECO
INST. FADER - GRAL. PACHECO
INST. DE LOS SGDOS. CORAZONES - HAEDO
COL. SHOLEM ALEIJEM - HAEDO
E.N.E.T. Nº 5 - HURLINGHAM
CTRO. COMUNITARIO Nº 5 - ISLA MACIEL
ESC. EDUC. MEDIA Nº 7 - I. CASANOVA
ESC. CRISTIANA EVANGELICA - ITUZAINGO
INST. PRIV. A. LINCOLN - ITUZAINGO
E.N.E.T. Nº 1 - JOSE C. PAZ
INST. GRAL. J. DE SAN MARTIN - JOSE C. PAZ
ESC. DE EDUC. MEDIA Nº 2 - JUNIN
INST. SUP. DE FORM. DOC. Nº 20 - JUNIN
COLEGIO MARIANISTA - JUNIN
ESC. DE CADETES GRAL. IRAMAIN - LA PLATA
INST. ANTONIO PROBOLO - LA PLATA
ESC. ENSEÑANZA MEDIA Nº 1 - LA PLATA
FAC. CIENCIAS VETERNARIAS - LA PLATA
FAC. CS. NATURALES - LA PLATA
INST. INV. BIOQUIMICAS - LA PLATA
ESC. CONCILIO VATICANO II - LA PLATA
COL. MARIA AUXILIADORA - LA PLATA
UNIV. NAC. DE LA PLATA - LA PLATA
INSTITUTO ATENEA - LANUS
INST. ECLESTON - LANUS
ESCUELA Nº 69 - LANUS
ESCUELA Nº 54 - LANUS
U.T.N. PACHECO - LOS POLVORINES
COL. SAN AGUSTIN - M. DEL PLATA
ESCUELA Nº 3 - MAR DEL PLATA
ESCUELA Nº 14 - MAR DEL PLATA
FUNDACION BOLSA DE COMERCIO - MAR DEL PLATA
CTRO. NAC. ENS. INFORMATICA - M. DEL PLATA
COL. STELLA MARIS - MAR DEL PLATA
COL. ALBERTO SCHWEITZER - M. DEL PLATA
ESCUELA Nº 67 - MAR DEL PLATA
ESCUELA Nº 62 - MAR DEL PLATA
ESCUELA Nº 27 - MAR DEL PLATA
ESCUELA Nº 31 - MAR DEL PLATA
C.E.F.A. - MAP DEL PLATA
JARDIN DE INFANTES MIS MANITOS - MAR DEL PLATA
INST. SUP. DE EST. ADMINISTRATIVOS - MAR DEL PLATA
INST. SAN VNTE. DE PAUL - M. DEL PLATA
JARDIN DE INFANTES Nº 2 - M. DEL PLATA
ESC. Nº 1 D.F. SARMIENTO - M. DEL PLATA
INST. DON ORIONE - MAR DEL PLATA
ESC. MAR DEL PLATA - M. DEL PLATA
INST. FAST COMPUTACION - M. DEL PLATA
ESC. ENS. MEDIA Nº 5 - MARTINEZ
STA. TERESA DEL N. JESUS - MARTINEZ
ESC. EDUC. MEDIA Nº 2 - MAYOR BURATOVICH
ESCUELA MEDIA Nº 3 - MEDANOS
ESC. ENSEÑANZA MEDIA Nº 4 - MERLO
E.N.E.T. Nº 1 - MORENO
INST. SAINT THOMAS BECKET - MUNRO
ESC. Nº 14 H. IRIGOYEN - NECOCHEA
INST. ARG. DE IDIOMAS - NECOCHEA
ESCUELA Nº 42 - NECOCHEA
ESC. ARG. DANESA ALTA MIRA - NECOCHEA
ESCUELA Nº 17 - OLAVARRIA
COL. CTRO. CULTURAL ITALIANO - OLIVOS
COL. LA ASUNCION DE LA VIRGEN - OLIVOS
COLEGIO TARBUT - OLIVOS
INST. D.F. SARMIENTO - OTAMENDI
ESCUELA Nº 28 - PALOMAR
INST. JOSE MANUEL ESTRADA - PELLEGRINI
AC. SUP. DE COMER. HELLER - PERGAMINO
INST/ CRISTO REY - QUILMES OESTE
COL. SANTO DOMINGO - RAMOS MEJIA
ESC. ARGENTINA DEL OESTE - R. MEJIA
INS. DE ENS. SUPERIOR - RAMOS MEJIA
INST. COMER. RANCAGUA - RANCAGUA
ESCUELA Nº 16 - REMEDIOS DE ESCALADA
ESCUELA Nº 30 - SALADILLO
COL. SAN FERNANDO - SAN FERNANADO
NTRA. SRA. DE LA UNIDAD - SAN ISIDRO
COL. CARDENAL SPINOLA - SAN ISIDRO
ESC. Nº 1 DR. COSME BECCAR - SAN ISIDRO
ESCUELA Nº 22 - SAN ISIDRO
INST. 20 DE JUNIO - SAN ISIDRO
CENTRO DE ESTUDIOS - SAN MARTIN
INST. NTRA. SRA. DE FATIMA - S. MANUEL
INST. SUP. DE FORM. DOCENTE Nº 42 - SAN MIGUEL
ESCUELA JUANA MANSO - SAN MIGUEL
COL. PARROQUIAL STA. N. TRUJUI - SAN MIGUEL
INST. SAN NICOLAS DE BARI - S. NICOLAS
E.N.E.T. Nº 1 - TANDIL
COLEGIO ECLESTON - TEMPERLEY
ESC. Nº 6 BME. MITRE - TIGRE
COLEGIO SAN RAMOS - TIGRE
ESC. AGROPECUARIA - TRES ARROYOS
E.N.E.T. Nº 1 - TRES ARROYOS
INST. JESUS ADOLESCENTE - 3 ARROYOS
E.N.E.T. Nº 1 - TRES ARROYOS
ESC. NAC. DE COM. M. BELGRANO - TRENQUE LAUQUEN
ESCUELA Nº 8 - TRENQUE LAUQUEN
ESCUELA Nº 17 - TRENQUE LAUQUEN
ESCUELA Nº 2 - TRENQUE LAUQUEN
E.N.E.T. Nº 1 - TRENQUE LAUQUEN
ESC. Nº 5 C. VILLEGAS - TRENQUE LAUQUEN
ESC. EDUC. MEDIA Nº 2 - VERONICA
INST. NUEVA ENSEÑANZA - VNTE. LOPEZ
INST. MIGUEL HAM- VICENTE LOPEZ
ESC. NAC. DE COM. M BELGRANO - VILLA BALLESTER
INST. NTRA. SRA. DE LOURDES - V. MADERO
E.N.E.T. Nº 1 J. NEWBERY - V. LUZURIAGA
INST. TEC. ALIMENTARIA - 9 DE JULIO
E.N.E.T. Nº 1 - 9 DE JULIO
ESC. MEDIA Nº 1 - 25 DE MAYO
E.N.E.T. Nº 1 - 25 DE MAYO

CHUBUT:
CTRO. PRIV. COMPUTACION EDUC. - TRELEW

CORDOBA:
COL. WILLIAM C. MORRIS - CORDOBA
COL. JESUS MARIA - LOS NARANJOS
INST. DE ENS. SUPERIOR - RIO CUARTO
CONVENTO DE SAN FSCO. - RIO CUARTO
COL. SAN BUENAVENTURA - RIO CUARTO
INST. DE 2º ENS. M. BELGRANO - SACANTA
S.E.S.F. COMPUTACION - SAN FRANCISCO
INST . JOSE PEÑA - VILLA CABRERA

CORRIENTES
TALLER GALILEO GALILEI - CORRIENTES
ESC. N. S. M. MANTILLA - CORRIENTES
INST. LOOK AND LEARN - CORRIENTES

ENTRE RIOS
U.T.N. - C. DEL URUGUAY
CTRO. C. I. Y DE LA PRODUCCION - C. DEL URUGUAY
E.N.E.T. Nº 2 - GUALEGUAY
FACULTAD DE BIOINGENIERIA - PARANA
ESC. INF ENTRE RIOS - PARANA
E.N.E.T. Nº 1 - PARANA
U.T.N. PARANA - PARANA
ENET Nº 1 PASCUAL ECHAGÜE - PARANA

JUJUY
ESCUELA J. I. GORRITI - S. S. DE JUJUY

LA RIOJA
INST. ARG. DE E. SECUNDARIOS - LA RIOJA
ESCUELA GABRIELA MISTRAL - LA RIOJA

MENDOZA
ESC. NAC. DE COM. M. BELGRANO - GODOY CRUZ
CENTRO INF. COMP. EDUCATIVA - MAIPU
INST. PADRE VASQUEZ - MAIPU
COL. VIR. DEL CARMEN DE CUYO - MAIPU
UNIVERSIDAD DE MENDOZA - MENDOZA
ESC. DE COMER. M. ZAPATA - MENDOZA
INSTITUTO PRAXIS - MENDOZA
INST. TECN. PRIV. T. EDISON - MENDOZA
INTERFACE CTRO. DE CAPACITACION - MENDOZA
INST. COMERCIAL PIO X - TUNUYAN

MISIONES:
S.M. DE PROM. DE LAS CIENCIAS - POSADAS
TALLER DE COMP. LAMPARITA - POSADAS

NEUQUEN:
ESCUELA Nº 11 - NEUQUEN
JAR. DE INFANTES PIMPINELA - NEUQUEN
ACT. G. ING. Y ARQUITECTURA - NEUQUEN
ESC. ENS. MEDIA Nº 32 - P. DEL AGUILA
CTRO. PROV. ENS. MEDIA Nº 3 - ZAPALA
CTRO. PRIV. DE INFORMATICA - ZAPALA

RIO NEGRO:
ESC. COMUN Nº 95 - GRAL. ROCA
ESC. Nº 168 FCO. RIVAL - GRAL. ROCA
ESC. COM. I. MALVINAS - GRAL. ROCA
COL. SECUNDARIO Nº 9 - GRAL. ROCA
E.N.E.T. Nº 1 - GRAL. ROCA
JAR. DE INFANTES PAYASIN - GRAL. ROCA
I.P.E.A. - GRAL. ROCA
ESC. Nº 71 S. MARTIN - S.C. DE BARILOCHE
INT. C.E.D.E.I. - S.C. DE BARILOCHE
COL. SEC. Nº 11 - VILLA REGINA
INST. NTRA. SRA. DEL ROSARIO - V. REGINA

SALTA:
INST. VACH - SALTA

SAN JUAN:
INSTITUTO BIOINGENIERIA - SAN JUAN

SAN LUIS:
INST. INFANTIL STA. CATALINA - SAN LUIS
INST. CAUSAY - SAN LUIS
INST. MASTERSOFT - SAN LUIS

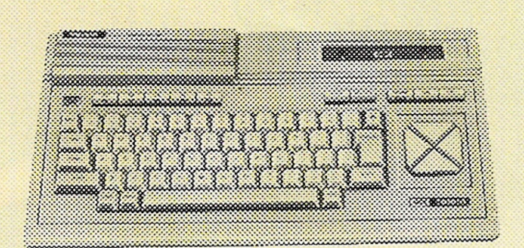
SANTA CRUZ:
ESC. Nº 5 CAP. ONETO - PTO. DESEADO
COL. SEC. Nº 8 NACIONES UNIDAS - PTO. DE STA. CRUZ

SANTA FE:
COLEGIO DE LOS ARROYOS - ROSARIO
INST. POLIT. SAN MARTIN - ROSARIO
SERVIRAMA - ROSARIO
COL. NAC. SAN LORENZO - ROSARIO
INST. NTRA. SRA. DE GUADALUPE ROSARIO
MAGIC COMPUTACION - ROSARIO
COL. SALECIANO S. JOSE - ROSARIO
ESC. Nº 55 D.F. SARMIENTO - ROSARIO
E.N.E.T. Nº 638 - ROSARIO
E.N.E.T. Nº 623 - ROSARIO
COLEGIO CRISTO REY - ROSARIO
ESC. COMP. LICEO RO-NES - ROSARIO
INST. SAGRADO CORAZON - SAN JORGE
INSTITUTO CORDOBA - SANTA FE
ESC. EDUC. TECNICA Nº 2 - SANTA FE
ENET Nº 2 - SANTA FE
UNIV. NAC. DEL LITORAL - SANTA FE
ESC. EDUC. TECNICA Nº 2 - SANTA FE
INST. PRIV. I. DEL N. JESUS - SAN JUSTO
INST. SUP. Nº 21 - SANTA FE

TUCUMAN:
INST. ING. SCHUAB Y TAPIA - TUCUMAN

Llene con sus datos el cupón al pie, envielo por correo y rebirá en forma GRATUITA la revista INFORMATICA Y EDUCACION.

TELEMATICA S.A.
CHILE 1347 -
(1098) CAPITAL FEDERAL
TEL.: 37-0051/4



Talent
Tecnologia y Talento
en el colegio

Sres. TELEMATICA S.A.
Chile 1347 - (1098) Capital Federal
Nombre
Cargo
Establecimiento Educacional
Dirección
Provincia

Leading

STAFF

LOAD MSX

Director General
Ernesto del Castillo

Director Editorial
Cristian Pusso

Director Periodístico
Fernando Flores

Secretario de Redacción
Ariel Testori

Arte y Diagramación
Fernando Amengual y Tamara Migelson

Departamento de Avisos
Oscar Devoto y Nelso Capello

Servicios Fotográficos
Image Bank, Oscar Burriel, Víctor Grubicy y Eduardo Comesaña

Load Revista para usuarios de la norma MSX es una publicación mensual editada por Editorial PROEDI S.A., Paraná 720, 5° Piso, (1017) Buenos Aires. Tel.: 46-2886 y 49-7130. Radiollamada: Tel.: 311-0056 y 312-6383, código 5941. Registro Nacional de la Propiedad Intelectual: E. T. M. Registrada. Queda hecho el depósito que indica la Ley 11.723 de la Propiedad Intelectual.

ISSN 0326-8241

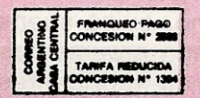
Precio de este ejemplar: ₳ 5

Impresión: Calcotam, Fotocromo tapa: Columbia. Fotocomposición: Interamericana Gráfica.

Los ejemplares atrasados se venderán al precio del último número en circulación.

Distribuidor en Capital: Martino, Juan de Garay 358, P. B. Capital.

Distribuidor interior: D G P: Hipólito Yrigoyen 1450, Capital Federal. T.E. 38-9266/9800.



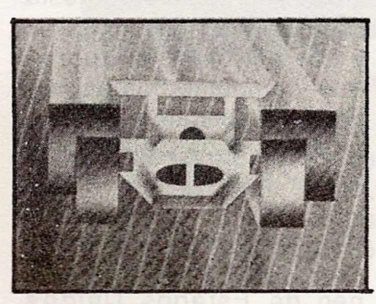
SUMARIO

CONCURSOS



Se entregaron los premios del 2do. Concurso de Programas, y entrevistamos a Eduardo Blotta, ganador del 2do. premio. Además, anunciamos al ganador del concurso mensual "Notas, Trucos y Programas", y el cierre del concurso trimestral. (Pág.**7**)

COMO ACELERAR UN PROGRAMA BASIC (1ra. parte)



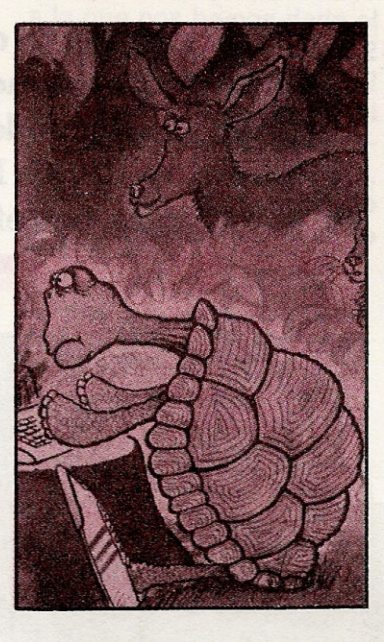
La velocidad de ejecución de los programas es importantísima para el usuario de microcomputadoras. En la carga de un juego o en los programas de comunicaciones, esta cualidad muchas veces es crítica. Para los que se enfrentan con este problema les brindamos las reglas que pueden solucionarlo. (Pág.**8**)

EL BASIC MSX Y EL DE PC



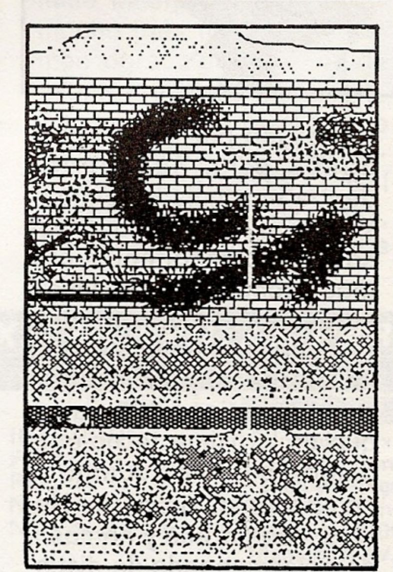
Nos referimos a las instrucciones iguales o de resultado similar, que permiten usar un programa de MSX en una PC. (Pág.**16**)

RINCON DEL USUARIO



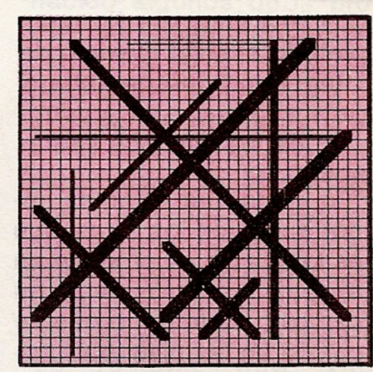
Explicamos el modo de encarar la enseñanza del Logo a través de una serie de preguntas-respuestas que clarifican y explican su alcance. (Pág.**18**)

LENGUAJE C PARA MSX: APLICACIONES



Presentamos el lenguaje C que está adquiriendo un gran auge en los círculos computacionales. Su velocidad y flexibilidad lo hacen ideal para la construcción de programas comerciales y sistema operativos. (Pág. **24**)

MANEJO DE PANTALLAS EN MODO 0



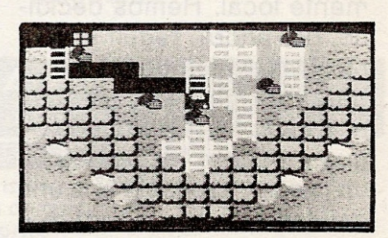
Para muchos, la norma MSX es la natural heredera de la TI-99/4A. Explicamos cómo a través de aquélla se han rescatado y mejorado la mayoría de sus cualidades como por ejemplo las facilidades para realizar gráficos desde BASIC en modo 2 de alta resolución. (Pág.**28**)

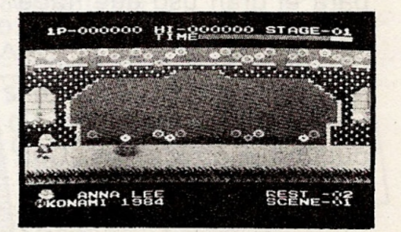
PROGRAMAS



Do-Re-Mi (Pág.12) -Scrash (Pág.20)

SECCIONES FIJAS





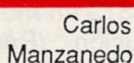
Noticias MSX (Pág.4) - Soft al día (Pág.31) - Buzón (Pág.34)

"LO UNICO QUE IMPORTAMOS SON LOS CHIPS"

Con el lanzamiento de la MSX2 se puso de manifiesto la autonomía de Telemática respecto a la dependencia de proveedores del exterior. Carlos Manzanedo, director de la empresa, se refiere al tema.

Carlos Manzanedo, director de **Telemática**, explicó que esa empresa ha completado el nivel máximo de integración que permitía el país al pasar a la producción de los circuitos impresos.

"Lo único que estamos importando son los chips, que es lo que no se produce en el país, -aclaró- pero, en cambio, parte de lo que se coloca en las plaquetas, como ser todos los componentes discretos, capacitores, resistores, etcétera, son provistos en forma local. El modulador también es fabricado localmente, así





Carlos Manzanedo

como la fuente, el transformador, la carcaza con la aparición de la **MSX2** se entrega con matricería totalmente local. Hemos decidido, por ahora, seguir importando los teclados porque hasta el momento no hay ninguna empresa en el país que produzca teclados, ni a un precio razonable ni a una calidad que sea aceptable para el gran uso que hace de ellos el público al cual va dirigido. Nosotros queremos ser muy respetuosos en cuanto a la calidad del producto que ofrecemos. Con las industrias que se van a radicar en los polos informáticos posiblemente se desarrolle una industria buena de teclados y ellos puedan ser proveedores nuestros."

-¿Y con respecto al diseño de la plaqueta?

-En el caso de la **MSX2**, es un producto totalmente desarrollado por **Telemática** desde cero. Se arrancó desde una hoja en blanco, se han seleccionado cuáles son los componentes que más tecnología han incorporado en el mercado. Esto se puede ver en la plaqueta en donde hay un componente que tiene 100 patas, que es la última tecnología disponible en cuanto a componentes de silicio. Hemos incorporado y hemos ido seleccionando de distintos proveedores del mercado, los componentes más convenientes para elaborar un producto con la máxima prestación y con el costo más bajo. Eso nos permite hoy poder producir una computadora sin tener ningún tipo de dependencia de proveedores del exterior.

Advirtió que hoy pueden seleccionar cuándo les conviene comprar en Japón, cuándo en Estados Unidos, y cuándo en cualquier otro país que ofrezca determinados componentes a un mejor precio.

O sea que realmente uno pierde los complejos en lo que respecta al proceso de desarrollo industrial -agregó-. En la fabricación de computadoras esto es posible. Tan posible que estamos a precios competitivos y nos permite entrar nuestras computadoras en mercados prácticamente libres como es el de Chile, donde los recargos para importación de computadoras no son más del 20%. Competimos con productos que vienen de Estados Unidos, Europa y Oriente, y no estamos teniendo problemas de precios .



MSX2 Turbo

-Generando divisas... era lo que se quería con el Plan Informático Argentino.

-Exacto. O sea, el objetivo del Plan Informático, que estaba apoyado en la Resolución 44 y después se cambió por la 652, era apoyar un fuerte desarrollo de la industria informática local que después permitiera encarar el mercado exportador. Lamentablemente, cuando salió esta resolución nosotros ya estábamos con nuestro propio proyecto muy avanzado en la provincia de San Luis por lo que no podíamos desarmar la planta y reeditarla en alguna de las zonas promocionadas. Pese a no tener los beneficios que otorgaba la 652 nosotros seguimos adelante con nuestras propias convicciones, con vocación industrial. Si bien en estos momentos se están inaugurando esos dos polos informáticos, con lo cual recién ahora tendríamos que empezar a pensar cómo hacer el proceso de integración, nosotros ya hemos hecho lo que pedía la 652: una integración máxima en dos años, ¡pero nosotros la logramos en uno! Cuando pedían el lanzamiento de un producto de tecnología propia que no significara ningún pago por transferencia de tecnología al exterior en un plazo de dos años, nosotros lo estamos haciendo en un año y medio. A su vez se exigía un plan exportador a partir de los dos años y nosotros también en menos de dos años ya empezamos a exportar. Y esto cumpliendo en un todo con lo que pide la 652 pero con una ventaja: cuando los que adhirieron a la 652 recién están empezando, nosotros ya nemos cubierto todas las etapas.

-Con respecto al service, a la parte técnica, ¿qué índice de fallas han detectado ?

-Bueno, quizá suene como que son argumentos nuestros, de fabricantes, pero

basta con consultarlo en el mercado. Se puede ver que nuestros productos realmente no tienen más de un 2% de fallas. Estamos sorprendidos frente a nuestras propias estimaciones. Recomendaciones originales de los que han desarrollado la **MSX1** estimaban un promedio de fallas del orden del 2 al 3% y nosotros creíamos que al pasar a la producción local podíamos estar entre el 3 y el 5%. A un año de tener la producción local, -nosotros entendemos por producir localmente no sólo poner cuatro tornillos, sino armar la plaqueta, proveer una plaqueta industria nacional- estamos manteniendo el mismo nivel de falla que cuando importábamos la plaqueta armada, en los primeros meses del año pasado.

-Y ustedes tienen servicio técnico propio.

-Nosotros tenemos una red de más de 40 centros de servicio técnico en todo el país. Son los que simplemente atienden fallas ocasionales, que a veces se producen por un maltrato dado a la máquina o a fallas mínimas, como lo demuestra ese 2% de fallas.

-Y bastante rápido ¿verdad?

-Sí, eso es parte de la consigna que dice que cuando no se puede arreglar algo en el momento, se busca hacer el reemplazo inmediato de la parte que puede estar fallada y después en laboratorio se busca con más detenimiento la falla.

-Esto ¿en el service central o en todos?

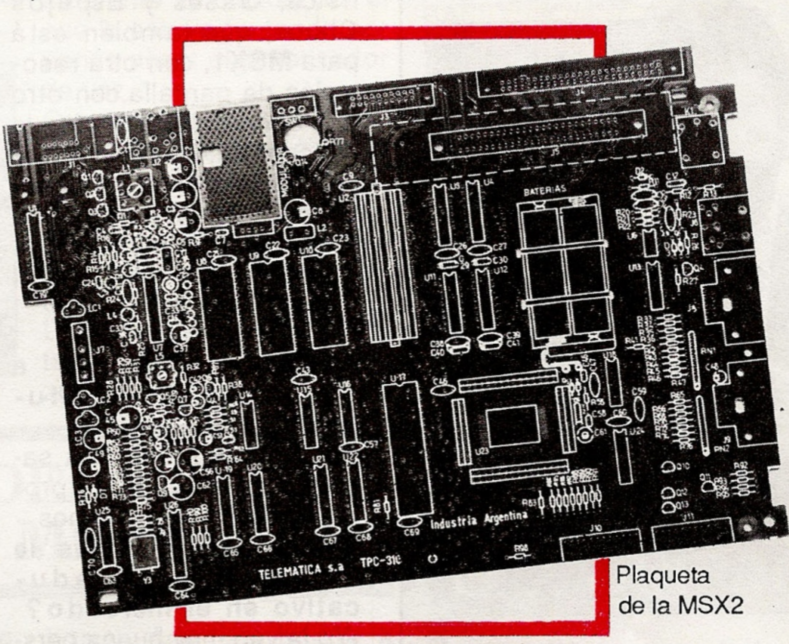
-Básicamente en casi todos; podemos decir que el 80, 90% de los servicios técnicos están trabajando de la misma manera.

-Que dicen los orientales, obviamente no los de Uruguay sino los de Oriente.

-Bueno, realmente los ha sorprendido mucho.

Nosotros estamos en trato permanente con los creadores de la norma MSX, que es la firma ASCII de Japón, y ellos están sorprendidos por el desarrollo tecnológico que se hace en Argentina. Tan sorprendidos que hay empresas del nivel de Philips, por ejemplo, u otras marcas, que ni siquiera tienen tecnología propia sino que compran productos a los que colocan su marca en otros países. Ellos están trabajando con gran cantidad de periféricos pero ninguno desarrollado bajo tecnología propia. Causa asombro que nosotros estemos ya sacando una **MSX2** a la que nos hemos dado el gusto de llamar **MSX2 Turbo** porque es la MSX2 más veloz que existe en el mundo. En cuanto a la selección de componentes, por haberlos producido prácticamente a 9 meses de diferencia con respecto al resto del mercado, nos permitió incorporar tecnología mucho más moderna y hemos logrado un mejor rendimiento y un más bajo costo. También los ha sorprendido cómo hemos hecho el desarrollo para la **MSX1** del cartucho de 80 columnas. No el desarrollo de otros cartuchos de 80 columnas que hay en el mercado sino que hemos entrado en un proceso de desarrollo y de miniaturización, porque ha habido algunos otros proveedores que han hecho cartuchos de 80 columnas pero con un tamaño que es el doble del nuestro y a un costo superior. Además desarrollamos la **Mini-Lan** que es la red local para uso educacional. Si bien existen otras redes educacionales en el mercado, están basadas en usar un equipo PC de 16 bits que vale muchísimo más que nuestro equipo central que es otra MSX cualquiera de las que está integrada a la red. Y sobre todo es la red local de uso educacional más económica que existe en el mundo.

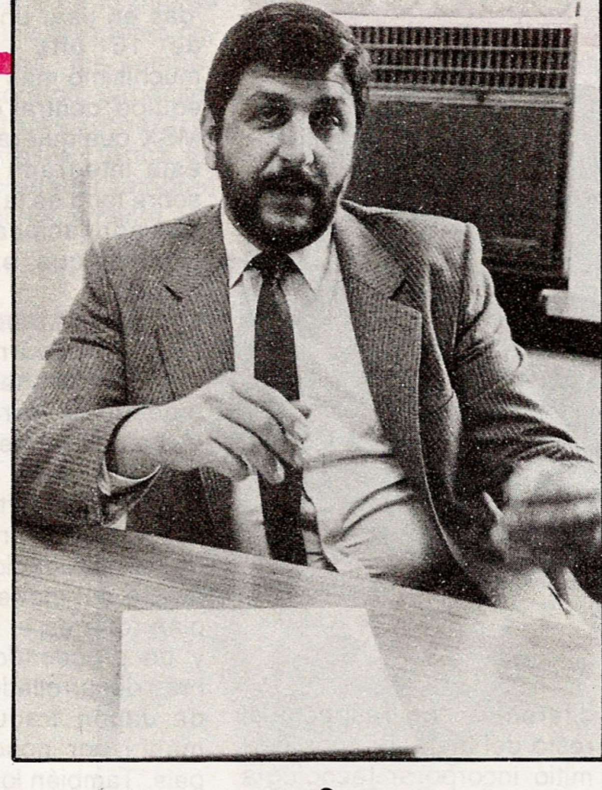
Los ha sorprendido muchísimo el desarrollo propio que hemos hecho de un modem de características únicas en el mercado mundial en cuanto a poder incorporarles soft inteligente como puede ser todo el soft de comunicación incluído dentro del modem, el Multiplan (que es el **MSX-Plan**) y un procesador de palabras desarrollado por ASCII de Japón traducido totalmente por nosotros en el país. También los sorprende que hayamos desarrollado el teclado numérico. Para eso contamos con nuestro Departamento de Ingeniería, con más de 15 ingenieros entre los que se dedican al área de software y a la de hardware.



Plaqueta de la MSX2

UNA EXCELENTE RESPUESTA

En la expansión del hardware producida en el mercado local, de la que se habla en esta misma página, el software educativo no ha quedado afuera. Para conocer características del incremento de su comercialización entrevistamos a **Antonio Giliberto**, director de Systemac.



Antonio Giliberto

¿Cómo fue la penetración de este software educativo?

-Bueno, la respuesta ha sido excelente.Si teníamos alguna duda al principio, realmente el usuario se está empezando a cansar del "mata marcianos", como yo los llamo. Quiere un software un poco más útil que aquel en el que simplemente toca un joystick, para disparar misiles y jugar. Realmente que sea más útil y que vaya formando al chico, para que empiece a entender desde una corta edad la importancia de la consola como una herramienta. Obviamente ningún software va a desarrollar la inteligencia del chico pero sí le va a empezar a introducir la filosofía de la computación que no se hace matando marcianos.

-¿Cuántos títulos tienen ya?

-Tenemos alrededor de 60 educativos ya, casi todos de procedencia europea (excepto dos que son desarrollo local), de firmas como **Dimensión New**, **Fhisher Price**, **Idealogic** y **Spinnaker**.

-¿Y los que son de desarrollo local, cuáles son ?

-Son uno de geografía y otro de matemáticas, muy curricular, que no han tenido muy buena acogida por ser muy estáticos. En cambio los otros son juegos educativos.

-Pero con respecto a los que son juegos educativos ¿ahí sí han tenido respuesta?

-Del docente y del alumno. Una excelente respuesta. Se han cuadruplicado las ventas en cinco meses.

-¿Responde a la currícula escolar ese software?

-No en forma directa, no es curricular, pero sí responde a cada materia específica. Por ejemplo, hay una serie que es **Logicolor** que es lógica metodológica y no sigue ninguna currícula de acá, pero se termina esa serie que son dos programas y se tiene muy en claro lo que es la lógica simbólica.

-¿Cuáles son los proyectos que tienen?

-Lanzaremos software educativo para la MSX2 fundamentalmente. Va a salir una serie de unos 25 títulos realmente educativos.

-¿Qué nos puede anticipar?

-Por ejemplo, había sobre física, Gases y Espejos Claros, que también está para **MSX1**, con otra resolución de pantalla,con otro tipo de software. Si bien la concepción, es decir la materia es la misma, es un poco más profundo porque la herramienta lo permite.

-¿También son del mismo origen?

-Sí.

-¿Para MSX1 van a seguir saliendo títulos?

-Sí. Nosotros estamos sacando al mercado un promedio de 8 títulos por mes.

-¿Qué perspectivas le ven al software educativo en el mercado?

-Yo le veo muy buena perspectiva. Día a día se venden más educativos y no pasa lo mismo con la venta de juegos, que está estática.

-En cuanto al desarrollo local ¿pensaron encarar algo nuevo?

-Sí, hay un par de casas de software que están trabajando. Estamos esperando que lo terminen para poder revisarlo y ver si cumplen con las expectativas que nosotros tenemos.



-Y en relación con el software de aplicación ¿en que están trabajando?

-Estamos trabajando con un paquete de gestión ventas, contabilidad, sueldos, video-clubes, administración de propiedades; se está terminando un paquete para inmobiliarias. Todos desarrollos locales. Evidentemente el soft de gestión tiene que ser de desarrollo local.

PARA VIDEO CLUBES

Un software diseñado y desarrollado para manejar toda gestión de un video club tipo es distribuido por **Organización Maro**. Ideal para cualquier computadora de norma MSX, el sistema está pensado para crecer junto con la empresa. Así, la versión estándar, que maneja **1500 títulos** y **500 clientes**, con el agregado de un disquete podrá ampliarse a **4000 títulos** y **1300 clientes**. Además de las posibilidades de realizar estadísticas, mailing y seguimineto de películas y clientes, que cubre la actividad del club en los últimos seis meses, el sistema ofrece una serie de características para el mejor funcionamiento del negocio, como por ejemplo: altas, bajas, consultas y modificaciones de películas, listados de títulos por géneros, control y listado de reservas, de deudores de películas y facturas, etcétera. (ver página 33).

Por otra parte, en lo que se refiere a sistemas generales, **Organización Maro** ofrece Contabilidad General, Stock y Cuentas Corrientes; y en el rubro de sistemas específicos presenta Administración Odontológica y Consultorios Médicos.

EL ROSTRO DEL SINTETIZADOR DE VOZ

LOS GANADORES

Telemática S.A. entregó los premios a los ganadores del 2do. Concurso de Programas. Manuel Rojas, quien resultó vencedor con su programa Do-Re-Mi, ya fue presentado en el número anterior. Esta vez nos visitó Eduardo Luis Blotta que con su Sintetizador de Voz (publicado en el número 14 , de Junio) obtuvo el segundo premio del concurso.

Eduardo es casi un ingeniero electrónico (le faltan rendir solo cuatro finales), que trabaja en la Facultad de Ingeniería de Mar del Plata haciendo desarrollos de hardware y software.

Participó en febrero de un curso de Microelectrónica dictado por EBAI (Escuela Brasilero Argentina de Informatica), en Tandil y en enero próximo asistirá a Curitiba a tres seminarios sobre Inteligencia artificial, sistemas expertos y control digital.


Manuel R Rojas

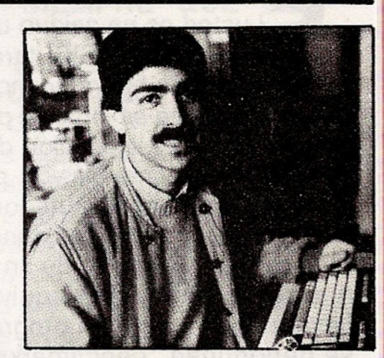
El programa se originó cuando comenzo a buscar un tema para el proyecto final de su carrera. Despuees de desechar algunos, finalmente se decidió por el que lo estuvo tentando por varios años: el sintetizador de voz.

Como etapa previa al desarrollo del hard, encaró el problema intentando realizar el sintetizador totalmente en software, de esa manera alcanzó un resultado muy interesante.

Un factor que tuvo en cuenta fue almacenarlo en una zona de memoria, que lo vuelva "transparente" al BASIC, es decir, que se inserta en la computadora con una instrucción más.

De visita en nuestra redacción, Eduardo se mostro entusiasmado por el lanzamiento de la MSX2, y anticipó su "salto" a esta máquina en un futuro cercano.

Sobre la utilidad del programa, nos comentó que, además, el proyecto, lo pensó con fines educativos, para acercar la informática a los no videntes, de manera que aquellos que no puedan ver la pantalla puedan en cambio escucharla.


Eduardo L. Blotta

GANADOR DEL CONCURSO MENSUAL

Fue **Miguel Luis Kopec** con su trabajo **Dibujando en la MSX**. La nota es una introducción al uso de isntrucciones gráficas. Presenta una serie de programas que permiten dibujar con línea llena, punteada y borrar. Además, brinda la posibilidad de variar las funciones cambiando algunos números o instrucciones dentro de los programas. El trabajo se publicará en el próximo número.

CIERRE DEL CONCURSO LOAD MSX

Recordamos a los que quieran participar del Concurso de notas, Trucos y Programas que el 15 de noviembre próximo finaliza el certamen. Por lo tanto, en el número de noviembre se conocerá el segundo ganador del concurso mensual, y el 15 de diciembre el ganador del premio trimestral.

Para quienes todavía no han participado, los requisitos son que los programas o trucos deberán servir para cualquier computadora de la norma MSX.

El **premio mensual** es un programa provisto por Telemática, a elección entre **Idea Base**, **Idea Text** o **Basic Tutor**.

El **premio trimestral** es un software de aplicación provisto por Telemática con un manual pudiendo optar entre el **MSX-LOGO**, **MSX-PLAN** o **MSX-WRITE**.

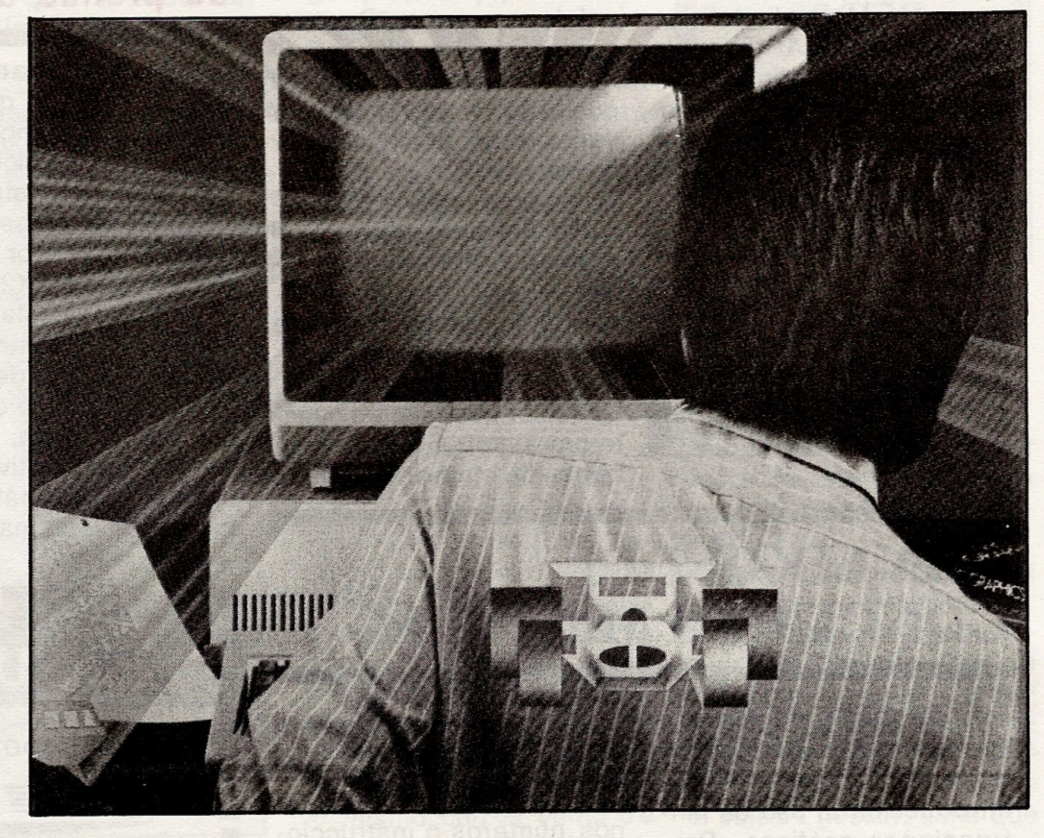
Los trabajos se deberán enviar a nombre de CONCURSO MENSUAL LOAD MSX, Paraná 720, 5º Piso, (1017) Cap. Fed.

COMO ACELERAR UN PROGRAMA BASIC - PARTE I

Hugo D. Caro

Es necesario agilizar la velocidad para que el software funcione de manera óptima. En la carga de juegos o en programas de comunicaciones, esta cualidad es imprescindible.

Si usted es un asiduo usuario del intérprete BASIC, tarde o temprano se habrá hecho esta pregunta: ¿Por qué es lento mi programa? La velocidad de ejecución de los programas es un tema clásico para los usuarios de microcomputadoras (como la MSX). El proceso que consiste en la medición de velocidad de un programa o sistema versus otro equivalente se lleva a cabo mediante programas que se denominan "benchmarks" o pruebas de escritorio. En la jerga informática de nuestro país se denomina "prueba de escritorio" aquella que se realiza con lápiz y papel, así que dejaremos el término yanqui "benchmark". Este proceso involucra la ejecución con tomas de tiempo de dos programas similares funcionalmente y la verificación de las diferencias de tiempo de corrida. Es una herramienta valiosa para comparar dos computadoras, pero no sirve para mucho más.



En este artículo discutiremos diversas reglas que la práctica en el uso de nuestro bienamado intréprete BASIC nos ha dictado para acelerar el funcionamiento de los programas, y para verificar la diferencias de tiempo de ejecución usaremos un programita sencillo que iremos modificando hasta lograr una reducción interesante.

Los tiempos y ejemplos se han aplicado sobre el intérprete MSX-BASIC y sobre el MBASIC o BASIC 80 versión 5.21, que es la versión de BASIC que acompaña al compilador BASIC 80 disponible en plaza para nuestra norma. Sin embargo, aquel usuario que se tope con nuestras líneas podrá hacer uso de las reglas aquí brindadas ya que se pueden aplicar en general sobre cualquier intérprete BASIC.

Antes que nada quisiéramos aclarar que la velocidad de ejecución en un programa no siempre es crítica, y existen otros factores (por ejemplo, que sea de fácil lectura o que ahorre espacio de memoria) que son más importantes. Sin embargo, se dan situaciones donde realmente es necesario que la velocidad de ejecución sea el punto de optimización del programa. Por ejemplo, los juegos tienen como problema principal tratar de reducir al máximo el tiempo de ejecución para lograr un efecto óptimo (es muy deprimente haber acertado un balazo al marcianito de siempre... y que la computadora no se dé por aludida). Asimismo, en los programas de comunicaciones (como los que traen el Modem y la Interfase RS-232 de Talent) también la velocidad es crítica, pues ésta es necesaria para que no se pierdan datos en el camino.

Antes de comenzar a sumergirnos en los mares de la velocidad de ejecución, veamos qué sucede cuando el usuario pulsa RUN si un programa sencillo, como por ejemplo el Programa 1, se encuentra en la memoria de nuestra Talent MSX:

Programa 1

```
10  LET N=0
15  REM *** COMIENZA EL
    CICLO
20  LET N=N+1
25  IF INT (N/7) = N/7
    THEN PRINT N;
30  REM ESTE CICLO VE
    RIFICA A CADA NU
    MERO MENOR QUE
    1000
35  REM PARA VER SI ES
    DIVISIBLE POR 7
40  REM SI ES ASI, SE
    IMPRIME EL NUMERO
45  IF N<1000 THEN GOTO
    20
50  REM *** FIN DEL CI
    CLO
```

La línea 10 está escrita en BASIC, por supuesto.

Muchos humanos pueden llegar a entender este lenguaje (si saben un poco de inglés), pero no es la lengua madre (lenguaje de máquina) de una microcomputadora. Antes de poder ejecutar la instrucción de la línea10, la computadora debe poder obtener las instrucciones equivalentes en su lenguaje nativo. Más que investigar todos los detalles del BASIC interpretado, po-

demos imaginar la ejecución de una línea o sentencia BASIC como un proceso de dos pasos, a saber:

1. Traducir la sentencia BASIC a código de máquina.

2. Ejecutar la sentencia en código de máquina.

La mayor desventaja de las versiones interpretadas de BASIC es que el paso 1. consume demasiado tiempo. Algunas líneas, como por ejemplo, las que se encuentran dentro de ciclos, deben traducirse muchas veces cuando se ejecuta un programa. Veamos ahora que podemos hacer con nuestro Programa 1 para acelerarlo. El programa no cumple una tarea muy excitante que digamos -tan solo imprime los números menores que 1000 y además son múltiplos de 7-. Es un programa corto (excluyendo los REM's tiene sólo 4 líneas). Sin embargo, podremos ver que aun siendo un programa tan simple existen muchas técnicas que acelerarán considerablemente la ejecución del mismo.

Es bastante razonable suponer que nuestros esfuerzos para reducir el tiempo de ejecución deben centrarse sobre los ciclos, porque son los que más tiempo consumen cuando se ejecuta un programa. Esto nos sugiere que usemos nuestras lupas sobre las líneas 20 a 45.

Las sentencias REM dentro de este ciclo debe haberle llamado inmediatamente la atención. La mayoría de nosotros sabe (aún con cierta reticencia) que los REM's o comentarios son necesarios para una mayor legibilidad (vaya palabra) de un programa. Sin embargo, cuando se encuentran dentro de un ciclo, pueden retardar la ejecución de un programa en varios órdenes de magnitud. Si los comentarios deben estar incluidos en los listados definitivos de sus programas, asegúrense de que no se encuentren en el cuerpo de un ciclo. Atención: hemos encontrado nuestra primera "regla sagrada de la velocidad":

REGLA 1: ELIMINE TODOS LOS COMENTARIOS (REM'S) DE LOS CICLOS DEL PROGRAMA

Cuando aplicamos nuestra primera regla sacrosanta sobre el Programa 1, obtenemos la nueva versión del mismo, que llamaremos Programa 1.1. Este nuevo programa corre bastante más rápido que su predecesor, en el caso del MBASIC hasta un 28 % más rápidamente. Las sentencias REM de las líneas 15 y 45 no han sido eliminadas pues no ahorrarían mucho tiempo, al estar fuera del ciclo. Tampoco es necesario eliminar la inicialización de las variables, ya que no se verifica ninguna disminución significativa.

Programa 1.1

```
10  LET N=0
15  REM *** COMIENZA EL
    CICLO 20 LET N=N+1
30  IF INT (N/7) = N/7
    THEN PRINT N;
40  IF N<1000 THEN GOTO
    20
45  REM *** FIN DEL CI
    CLO
```

En general, ¿cuánto tiempo se ahorra realmente cuando se eliminan los REM's? Naturalmente, toda reducción de tiempo depende del número de comentarios que incluya el ciclo y las veces que se ejecuta el mismo. Pero también influye la longitud de las sentencias REM. No es lo mismo la "ejecución" de una sentencia REM con 3 caracteres que con 70...

En la mayoría de las versiones de BASIC, los ciclos pueden construirse de dos formas: con sentencias GOTO, y con sentencias FOR-NEXT (en el caso del MBASIC, también la sentencia WHILE permite, junto con WEND, crear ciclos).

En el Programa 1 hemos utilizado la sentencia GOTO, pero es obvio que se podría haber utilizado un ciclo FOR-NEXT de la siguiente manera:

Programa 1.2

```
20 FOR N=1 TO 1000
30 IF INT (N/7) = N/7
   THEN PRINT N;
40 NEXT N
```

Nótese que este programa es un poco más corto que el Programa 1 y que corre bastante más rápido. De hecho, en MSX-BASIC corre un15% más rápido. Naturalmente, cuando se mejora la velocidad de ejecución de un programa se puede perder a veces un poco de la legibilidad del mismo. Sin embargo, en este caso no se aplica, ya que siempre se puede visualizar mejor un ciclo construido con FOR-NEXT que con GOTO. Llegamos pues a la segunda "regla sagrada":

REGLA 2: SIEMPRE QUE SEA POSIBLE, UTILICE CICLOS FOR-NEXT EN VEZ DE CICLOS CONSTRUIDOS CON SENTENCIAS GOTO.

Para ver más claramente la diferencia de tiempos, es mejor comparar la velocidad de ejecución de ciclos "vacíos". Esta medición tomará el tiempo de ejecución de la estructura en sí misma. Ciclo 1 y Ciclo 2 han sido tomados de los programas Programa 1 y Programa 2. La tabla 1 compara las velocidades de ejecución de ambos.

Ciclo 1

```
10 N=0
15 N=N+1
20 IF N<1000 GOTO 15
```

Ciclo 2

```
10 FOR N=1 TO 1000
20 NEXT N
```

TABLA 1:

Comparando ciclos vacíos

Programa	MBASIC	MSX-BASIC
Ciclo 1	6.44 s	6.00 s
Ciclo 2	2.44 s	2.25 s

Cuando se utilizan GOTO condicionales se obtiene más velocidad que cuando se utilizan los GOTO incondicionales. Por ejemplo, el siguiente ciclo utiliza un salto incondicional a la línea 30 y requiere un poco más de tiempo de ejecución. También puede aplicarse este criterio a los ciclos WHILE-WEND que existen en el MBA-

SIC, pues los mismos corren un poco más lentamente que el Ciclo1, por lo tanto la Regla 2 también se aplica a los ciclos WHILE-WEND de la misma forma que con los ciclos creados con sentencias GOTO.

Ciclo 3

```
10  N=N+1
20  IF  N>1000  GOTO  40
30  GOTO  10
40  REM EL PROGRAMA
    CONTINUA  AQUI...
```

Podemos aumentar aún más el tiempo de ejecución si eliminamos la N de la sentencia NEXT N, y así obtenemos al Ciclo 4. Cuando la sentencia NEXT se refiere a una variable, se demora el tiempo de ejecución.

Ciclo 4

```
10  FOR N=1  TO  1000
20  NEXT
```

Al no especificar la variable N en el Ciclo 4, eliminamos la necesidad de que la computadora busque una variable, y disminuye el tiempo de ejecución. Los ahorros son suficientes como para justificar el dictado de la tercera "regla sagrada":

REGLA 3: LOS CICLOS FOR-NEXT SE EJECUTAN MAS RAPIDAMENTE CUANDO NO SE ESPECIFICA LA VARIABLE EN LA SENTENCIA NEXT

Si aplicamos esta regla a nuestro Programa 1.2, obtenemos cerca de un 3% de aumento en la velocidad. Más adelante veremos que si aplicamos esta regla a un programa más complejo, el ahorro de tiempo será más significativo.

Programa 1.3

```
20  FOR N=1  TO  1000
30  IF  INT  (N/7)  =  N/7
    THEN  PRINT  N;
40  NEXT
```

Hemos aplicado 3 reglas, y ahora nuestro programa sólo tiene 3 líneas de longitud. Sin embargo, ¡¡todavía pueden aplicarse dos reglas más!! La primera surge de la observación de que la división de la línea 30 se ejecuta dos veces. Es posible que el programa corra más rápidamente si encontramos una forma para que se ejecute sólo una vez. Pero esto es fácil: ¡guardemos el resultado en una variable! Cuando hacemos esto, obtenemos el Programa 1.4.

Programa 1.4

```
20  FOR  N=1  TO  1000
25  LET  CHECK=N/7
30  IF  INT(CHECK)=CHECK
    THEN  PRINT  N;
40  NEXT
```

En MBASIC, el aumento de velocidad es despreciable, pero sobre MSX-BASIC se logró un aumento del orden del 27%. Sin embargo, el programa es ahora más largo y tiene una variable adicional; hemos canjeado velocidad por espacio de memoria. Este trueque se aplica reiteradamente cuando se necesita acelerar la velocidad de trabajo de un programa. Decidir cuándo debe aceptarse este canje no es siempre fácil, pero de todos modos tenemos nuestra siguiente regla sagrada:

REGLA 4: SI EL MISMO CALCULO SE EFECTUA MAS DE UNA VEZ, EL TIEMPO DE EJECUCION PUEDE ABREVIARSE A VECES UTILIZANDO UNA VARIABLE PARA ALMACENAR EL RESULTADO Y LUEGO APLICARLA EN LA FORMULA, EN VEZ DE REPETIR EL CALCULO.



Al aplicar la Regla 4 al Programa 1.3, hemos podido eliminar un cálculo, N/7, a cambio de agregar una sentencia de asignación. Dado que la división es una operación que consume bastante tiempo en el MSX-BASIC, hemos obtenido un aumento significativo en el tiempo de ejecución. En cambio, en MBASIC, la división es un poco más eficiente y además se trabaja con números de simple precisión, y por lo tanto nuestro reemplazo no ahorra mucho tiempo.

La Regla 4 trabaja mejor cuando se eliminan más de dos cálculos. El reemplazo de una operación más rápida, como por ejemplo "+", debe hacerse con cuidado -puede ser necesaria cierta experimentación-. El tiempo de ejecución de un cálculo también depende de que sus operandos sean constantes o variables. En algunas versiones de BASIC se puede ahorrar tiempo restringiendo el uso de constantes tales como 7 o 3.14159. Para implementar esta restricción podemos utilizar variables tales como S o PI. Luego, cada vez que se necesite utilizar una constante, se referencia mediante la variable apropiada. Esta es la idea que trasunta detrás de la Regla 5, que es la primera regla que no trabaja igual para MBASIC que para MSX-BASIC (en MBASIC no funciona):

REGLA 5: SI UTILIZA MUCHAS VECES UNA CONSTANTE DENTRO DE UN PROGRAMA, ENTONCES ALMACENE SU VALOR EN UNA VARIABLE Y LUEGO USE LA VARIABLE EN VEZ DE LA CONSTANTE.

Si aplicamos esta regla sobre el programa anterior, obtenemos:

Programa 1.5

```
10  LET  S=7
20  FOR  N=1  TO  1000
25  LET  CHECK=N/S
30  IF  INT(CHECK)=CHECK
    THEN  PRINT  N;
40  NEXT
```

La mejora de velocidad, aun en MSX-BASIC, no alcanza al 4%. Sin embargo, los ahorros obtenidos con esta regla pueden ser substanciales. Por ejemplo, si la aplicamos a los siguientes segmentos de programa (ver Tabla 2), encontramos que la Regla 5 es efectiva aun para el MBASIC. Ambos segmentos calculan el área de un círculo 1000 veces.

Antes de la Regla 5

```
30  FOR  N=1  TO  1000
40  LET  AREA  =  3.14159
    *  14  *  14  50  NEXT
```

Después de la Regla 5

```
10  LET  PI=3.14159
20  LET  R=14
30  FOR  N=1  TO  1000
40  LET  AREA=PI*R*R
50  NEXT
```

TABLA 2

Tiempos de ejecución en segundos antes y después de la Regla 5.

	MBASIC	MSX-BASIC
Antes	7.12	12.41
Después	6.63	10.35

Existe una aplicación levemente diferente para la Regla 5, para las variables alfanuméricas. Cuando uno hace la asignación de una constante tal como CHR$(12) por CLS$, es mucho más rápido acceder a la variable CLS$ que calcular CHR$(12). Esta aplicación de la Regla 5 funciona también sobre el MBASIC.

Por último queda una regla más que se puede aplicar para ahorrar tiempo de ejecución. Esta regla es ligeramente diferente a las otras en la filosofía de

aplicación. Se pueden llegar a obtener ahorros espectaculares de tiempo, pero posiblemente sea muy difícil o virtualmente imposible de aplicar.

La regla es:

REGLA 6: SI LAS REGLAS ANTERIORES NO ACELERAN LO SUFICIENTE A SU PROGRAMA, TRATE CON OTRO ALGORITMO

En el caso de nuestro querido programita, la tarea a realizar se puede enfocar desde otro ángulo. La observación clave que se aplica aquí es que los números que son divisibles por 7 son precisamente aquellos que son múltiplos de 7 (por ejemplo, los números 7, 14, 21...). Podemos encontrar estos números por simple suma y evitar todas la verificación de números que sabemos que no son divisibles por 7. Generalmente, una nueva versión de un programa es significativamente más rápida que las versiones anteriores. El siguiente programa, por ejemplo, ¡¡corre al menos 23 segundos más rápidamente que la versión 1.1!!

Programa 1.6

```
10 LET S=7
15 REM *** COMIENZA EL
   CICLO
20 PRINT S;
30 LET S=S+7
40 IF S<1000 THEN 20
45 REM *** FIN DEL CI
   CLO
```

En vista de las reglas que hemos analizado, es evidente que el programa 1.6 no es la versión más rápida que se puede implementar del nuevo algoritmo. Se puso este ejemplo para demostrar que un algoritmo superior es normalmente mucho mejor que cualquier versión anterior de un algoritmo inferior. El siguiente programa también está basado en esta nueva versión, pero se utiliza un ciclo FOR-NEXT con la opción STEP, que lo hace considerablemente más rápido que el ciclo GOTO de antes:

Programa 1.7

```
10 FOR S=7 TO 1000
   STEP 7
20 PRINT S;
30 NEXT
```

Es realmente sorprendente que se le haya podido aplicar 6 reglas de reducción de tiempos a un programa tan simple como Programa 1. La tabla 3 resume las mejoras de tiempo que se logran con las distintas versiones al aplicar las 6 reglas.

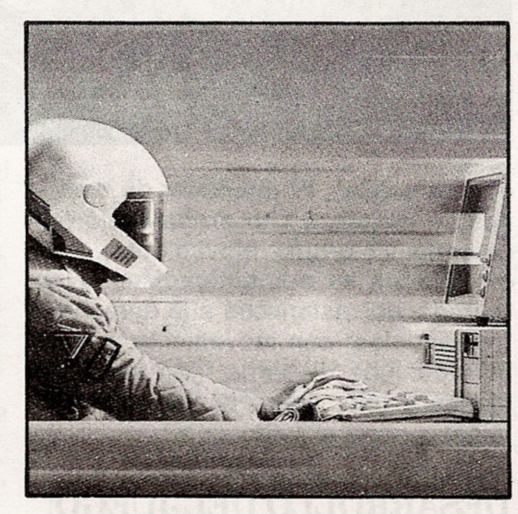


TABLA 3
Tiempos de ejecución después de aplicar reglas 1-6
(Ver nota al pie)

Regla	MBASIC	MSX-BASIC
Sin reglas	26.69	32.19
1	19.25	26.37
2	14.94	22.50
3	14.59	21.88
4	14.13	15.97
5	14.25	15.43
6	3.06	1.50
Programa 1.7	2.28	0.97

Aclaración: es notable la diferencia de velocidad a favor del MBASIC, pero se debe a varias circunstancias:

1. Como tipo por defecto utiliza variables simple precisión.

2. Se hicieron las pruebas sobre 80 columnas, lo que ahorra tiempo en la impresión.

Recordemos estas reglas cada vez que programemos, y encontraremos que se pueden podar varios segundos del tiempo de ejecución del programa. Continúen sintonizando esta estación (o sea compren los siguientes números de esta revista) ya que vendrán más chimentos sobre cómo acelerar la ejecución de sus programas.

NOTA IMPORTANTE:

En este artículo hemos asumido que Uds. desean escribir programas BASIC más eficientes que se ejecutarán sobre intérpretes. Una alternativa importante del BASIC interpretado es el BASIC compilado. Un compilador BASIC traduce el programa completo escrito en BASIC a código ejecutable de máquina antes de que el mismo se ejecute, y crea un módulo llamado de "run-time" o de tiempo de ejecución. Si se utiliza un compilador BASIC (como el que acompaña al MBASIC) la traducción de una línea de programa se efectúa una sola vez (durante la compilación del programa) para que cuando éste se ejecuta, no sea necesaria ninguna traducción adicional. Se deduce pues que un programa compilado ejecuta mucho más rápidamente que uno interpretado (ell MSX-BASIC no posee aún un compilasdor como el BASIC 80, pero hay noticias que indican que aparecerán en algún momento...).

Si se requiere todavía más velocidad, se puede lograr creando programas eficientes en lenguaje Assembler o ensamblador. Sin embargo, para muchas personas es suficiente con los métodos que aquí se han explicado, ya que no tienen un compilador BASIC y no les interesa o no saben programar en lenguaje Assembler.

Hugo D. Caro

DO RE MI

Clase: Entrenamiento musical
Autor: Manuel R. Rojas

(2da. Parte)

En el número anterior brindamos la presentación, descripción y partes de este programa. Además, en la lista de funciones llegamos hasta la elaboración de la segunda pantalla. En esta ocasión completamos el programa explicando el desarrollo del juego y cómo acceder a la tercera pantalla.

Con un disparo pasamos a la tercera pantalla, esta es la más simple. Elaborada sobre SCREEN 0, se utiliza para elegir el "Nivel" (entre 0 y 9) la cantidad de jugadores y la forma en que jugarán.

Una vez hecha la última elección (ellas pueden hacerse indistintamente desde el teclado o desde el joystick, fijándolas con un disparo) se pasa automáticamente a la cuarta y última pantalla. Aquí podemos visualizar el dibujo de un teclado completo de piano con dos cursores que se mueven independientemente y que sirven para acotar el intervalo de escala sobre el cual trabajaremos (se puede optar por cualquier longitud de "intervalo" ubicado a cualquier altura del teclado, aunque los cursores se presentan acotando una escala en la mitad del mismo). Debajo del "pianito" se ve un panel que marca dificultades para que nosotros elijamos su nivel:

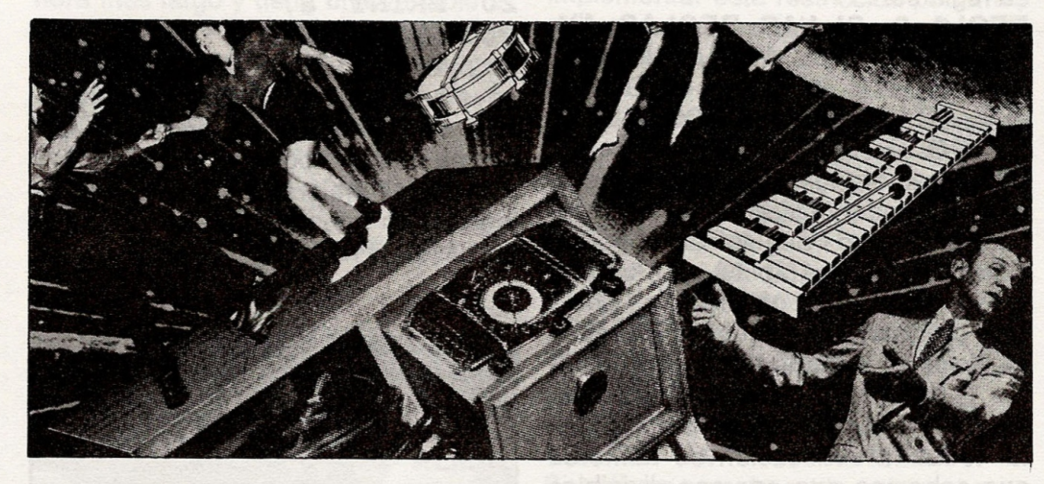
1) "Intervalo" de escala musical (de este "intervalo" el programa extraerá al azar las "notas" que deberemos reconocer). Este se define por la nota de comienzo (NC%-cursor izquierdo) y la nota final (NF%-cursor derecho). Ambos se desplazan con movimientos izquierda/derecha y se fijan con un disparo.

2) Cantidad de "Tentativas" que creemos necesitar para acertar nuestra elección (CT%).

3) Cantidad de "Notas" que nos creemos capaces de reconocer (CN%). (Estas difucultades, que selecciona el propio jugador, definen un "Premio" (PM%) o puntaje máximo que podemos llegar a obtener en nuestra jugada).

4) "Volumen" del sonido (VS%).

Un ícono móvil indica al jugador qué opción está haciendo. Esta elección puede ser corregida todas las veces que se desee antes de comenzar a jugar, ya que la corrección del valor del "premio" es automática y, en cierta forma, otro factor más de elección, ya que la opción de juego puede ser alcanzar determinado puntaje por combinación de distintos tipos de dificultades (esto puede ser útil jugando de a dos, cuando se conocen las flaquezas del oponente o se especula con el orden de salida). El marcador de opciones se desplaza con arriba/abajo (siguiendo su propio movimiento). Las dificultades se aumentan/disminuyen "agregando" asteriscos a la derecha o "retirándolos" con movimiento a la izquierda. Los disparos fijan las decisiones y uno, sobre la orden de comienzo, inicia el juego en sí.



DESARROLLO DEL JUEGO

Primero se oirá la escala completa correspondiente al "intervalo" previamente determinado. Simultáneamente aparece un "marcador" que va señalando tecla a tecla (incluyendo bemoles y sostenidos) la ubicación de cada sonido. Terminada la escala este "marcador" salta al medio. Se escuchan entonces la o las "notas" que deberán reconocerse. Estas se repetirán durante el desarrollo del juego. Se comienza por adivinar la primera, y así sucesivamente, moviendo el "marcador" (izquierda/derecha) hacia la tecla que creemos que corresponde. Con un disparo fijamos nuestra decisión. Una señal auditiva indicará el acierto (campanillas) o el error (ruido grave). En el primer caso desaparecerá un asterisco de "notas" y se eliminará su sonido, dejando oír solo las restantes. En ambos casos desaparecerá un asterisco en "tentativas" y así disminuirán nuestras posibilidades de elección De esta forma seguirá hasta adivinar todas las "notas" o gastar todas las "tentativas".

Durante el juego es posible reclamar una "ayuda" al programa, que consiste en la repetición de la emisión de la escala completa (incluyendo el "marcador" recorriendo el teclado). En este caso el "premio" se verá reducido en un factor de 0,1*NI% por cada ayuda.

Cuando el juego termina se otorga un puntaje valorativo del desempeño del jugador o los jugadores. Cuando este sea máximo (y no se haya pedido ninguna "ayuda") el programa saludará el desempeño del usuario con una marchita.

El juego con dos jugadores es similar y lo ejecutan ambos simultáneamente. El acierto de uno, cuando se logra sin pedir ayuda, es premiado y con la posibilidad de hacer la jugada siguiente. El jugador que está en turno se indica en un recuadro debajo del pianito y aparecen ahora Premio 1 y 2, y Puntaje 1 y 2 (obviamente, la cuarta pantalla posee ligeras variaciones, según jueguen uno o dos jugadores).

LISTA DE FUNCIONES DEFINIDAS POR EL USUARIO

Pantalla 3

K%: Cursor (0...1).
J%: Renglón del cursor.
NI%: Nivel de juego (0...9).
JU%: Cantidad y equipamiento de juadores (0...4).
I%: Compás del "Minué".

Pantalla 4

I%, J%, K%: Indices varios.
K%: Variable global: si K%=1, el juego recién comienza; si no, ya se ha jugado alguna vez.
NC%: Nota de comienzo del intervalo.
NF%: Nota del final del intervalo.
CT%: Cantidad de tentativas.
CN%: Cantidad de notas a adivinar.
VS%: Volumen del sonido.
L%: STICK.
JU%: Cantidad y equipamiento de jugadores:
1 Un jugador con teclado.
2 Un jugador con joystick.
3 Dos jugadores con joystick (juega 1).

4 Dos jugadores (con joystick y teclado) juega 1.
5 Idem 3 pero juega 2
6 Idem 4 pero juega 2
A1%: Ayuda para el jugador 1.
A2%: Idem jugador 2.
NO%: Nota sobre el teclado.
CO%: Señalador de opción oscila entre 0 y 1.
PM%: Premio cuando juega un solo jugador.
LI%: Límite izquierdo del intervalo.
LD%: Idem derecho.
N%: Color de los apuntadores oscila entre 0 y 1.
M%: Modelo del SPRITE de los límites (-1 o 0).
P%: Posición del límite que se va corriendo.
J%: Dirección a la que apunta el STICK.
S% (15): Notas al azar (0...95).
CA%: Cantidad de aciertos para un solo jugador.
CE%: Cantidad de errores ídem.
PP%: Puntaje.
MX%: Puntaje máximo de la sesión.
P1%, P2%, P3%, P4%: Premios 1, 2, 3, 4 respectivamente.
E1%, E2%, E3%, E4%: Errores 1, 2, 3, 4 respectivamente.
AY%: Ayuda.

```
2480 '
2490 '
2500 '          PANTALLA 3
2510 '       Elecci"n del nivel
2520 '
2530 COLOR15,4,4:SCREEN0:KEYOFF
2540 LOCATE5,3:PRINT"NIVEL (0..9): "
2550 LOCATE5,9:PRINT"1   UN JUGADOR CON TECLADO"
2560 LOCATE5,11:PRINT"2   UN JUGADOR CON JOYSTICK"
2570 LOCATE5,13:PRINT"3   DOS JUGADORES CON JOYSTICK"
2580 LOCATE5,15:PRINT"4   DOS JUGADORES CON JOY.Y TECL."
2590 K%=1:ONINTERVAL=7GOSUB2820:ONSTRIGGOSUB2840,2840,,2840,
2600 LOCATE19,3,K%:INTERVALON:STRIG(0)ON:STRIG(1)ON:STRIG(3)ON
2610 A$=INKEY$:IFA$=""THEN2630ELSEIFASC(A$)=13THENGOSUB2840
2620 IFASC(A$)>47ANDASC(A$)<58THENNI%=ASC(A$)-48:GOTO2650
2630 J%=STICK(0):IF0<J%THEN2640ELSEJ%=STICK(1):IF0=J%THEN2610
2640 IFJ%=1ORJ%=2ORJ%=3THENNI%=(1+NI%)MOD10ELSENI%=(NI%+9)MOD10
2650 LOCATE18,3:PRINTNI%:LOCATE19,3,K%:FORJ%=0TO1000:NEXT:GOTO2610
2660 '
2670 '
2680 'Equipamiento del o los jugadores
2690 '
2700 '
2710 J%=9:K%=1:ONINTERVAL=10GOSUB2850:ONSTRIGGOSUB2880,2880,,2880,
2720 STRIG(0)ON:STRIG(1)ON:STRIG(3)ON:INTERVALON
2730 LOCATE7,J%,K%:FORL%=0TO2900:NEXT
2740 A%=STICK(0)+3:IF0=(A%MOD4)THEN2760ELSEA%=STICK(1)+3:IF0=(A%MOD4)THEN2760
2750 A$=INKEY$:IFA$=""THEN2740ELSEIF13=ASC(A$)THENGOSUB2880ELSEJU%=ASC(A$)-48:IF
JU%>4ORJU%<1THEN2740ELSEJ%=7+2*JU%:GOTO2730
2760 J%=8+((J%+A%+2)MOD8):GOTO 2730
2770 '
2780
2790 '       Subrutinas para                    interval y strig
2800 '
2810 '
2820 K%=1-K%:L%=PEEK(63833!)-PEEK(63834!):IF0<=L%ANDL%<7THENGOSUB2870
2830 RETURN2600
2840 LOCATE18,3:PRINTNI%:INTERVALOFF:STRIG(0)OFF:STRIG(1)OFF:STRIG(3)OFF:RETURN2
710
2850 K%=1-K%:L%=PEEK(63833!)-PEEK(63834!):IF0<=L%ANDL%<7THENGOSUB2870
2860 INTERVALON:LOCATE7,J%,K%:RETURN2740
2870 I%=(I%+1)MOD40:PLAYMU$(0,I%),MU$(1,I%),MU$(2,I%):RETURN
2880 INTERVALOFF:STRIG(0)OFF:STRIG(1)OFF:STRIG(3)OFF:JU%=.5*(J%-7):GOSUB2870:RET
URN2940
2890 '
2900 '
2910 '          PANTALLA 4
2920 '  Sprites para los apuntadores
2930 '
2940 SCREEN2:COLOR15,4:OPEN"grp:"FOROUTPUTAS#1:I%=USR0(0)
2950 SPRITE$(0)=CHR$(8)+CHR$(8)+CHR$(8)+CHR$(20)+CHR$(34)+CHR$(34)+CHR$(20)+CHR$
(8)
2960 SPRITE$(1)=CHR$(8)+CHR$(8)+CHR$(8)+CHR$(24)+CHR$(24)+CHR$(24)+CHR$(56)+CHR$
(120)
2970 SPRITE$(2)=CHR$(8)+CHR$(8)+CHR$(8)+CHR$(12)+CHR$(12)+CHR$(12)+CHR$(14)+CHR$
(15)
2980 '
2990 '
3000 '          Pianito
```

CONCURSO

LOAD MSX

NOTAS, TRUCOS Y PROGRAMAS

AUSPICIADO POR TELEMATICA S.A. fabricante en Argentina de las computadoras personales TALENT MSX

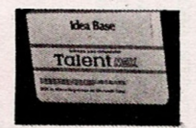
Se premiarán los mejores trabajos. Los programas y trucos deberán servir para cualquier computadora de la norma MSX. Las notas deben apuntar a aprovechar los recursos de los equipos.

MENSUAL — PREMIOS — TRIMESTRAL

MENSUAL: Un programa provisto por Telemática, a elección entre IDEA BASE, IDEA TEXT o BASIC TUTOR







TRIMESTRAL: Un software de aplicación, provisto porTelemática, con su manual pudiendo optar el ganador entre MSX LOGO, el MSX PLAN o el MSX WRITE.







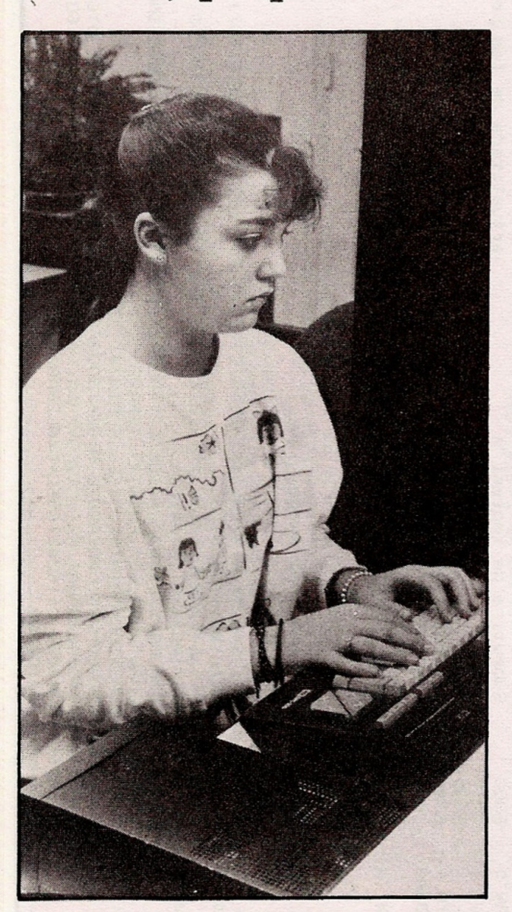
Pueden escribir a nombre de CONCURSO MENSUAL LOAD MSX Paraná 720, 5º Piso (1017) Cap. Fed.

```
3010
3020
3030 K%=BASE(10):FORJ%=0TO6:VPOKEK%+J%,7:NEXTJ%
3040 DATA255,157,201,221,220,136,127
3050 DATA1,2,3,2,4,1,4
3060 RESTORE3040:K%=BASE(12)
3070 FORI%=0TO6:READA%
3080 FORJ%=0TO7:VPOKEK%+J%,A%:NEXTJ%
3090 K%=K%+8:NEXTI%
3100 K%=BASE(11)
3110 FORJ%=0TO55:VPOKEK%+J%,31:NEXTJ%:FORJ%=4TO112STEP32:VPOKEK%-J%,7:NEXTJ%:LIN
E(0,64)-(255,151),12,BF
3120 K%=BASE(10)+34
3130 FORL%=0TO1:RESTORE3050
3140 FORI%=0TO6:READA%
3150 FORJ%=0TO21STEP7:VPOKEK%+J%,A%:NEXTJ%
3160 K%=K%+1:NEXTI%
3170 K%=K%+25:NEXTL%
3180 FORI%=0TO1
3190 FORJ%=0TO27:VPOKEK%+J%,5:NEXTJ%
3200 K%=K%+32:NEXTI%
3210 FORJ%=4TO112STEP32:VPOKEK%-J%,7:NEXTJ%:LINE(0,64)-(255,151),12,BF:I%=USR1(0
):INTERVALOFF:IFJU%>2THENLINE(17,51)-(87,62),1,B:DRAW"BM24,53":PRINT#1,"JUEGA 1"
3220 '
3230 '
3240 '   Selecci"n de dificultades
3250 '
3260 '
3270 K%=-1
3280 IFK%THENNC%=36:NF%=47:CT%=1:CN%=1:VS%=1:DRAW"bm32,72":PRINT#1,"LIMITES"
3290 L%=(JU%-1)MOD5:L%=-(L%>0)-(L%>3):A1%=0:GOSUB3630'Cambiar los l/mites en el
pianito
3300 IFK%THENDRAW"bm32,88":PRINT#1,"TENTATIVAS"
3310 GOSUB3860'Cambiar la cantidad de tentativas
3320 IFK%THENDRAW"bm32,104":PRINT#1,"NOTAS"
3330 GOSUB4010'Cambiar la cantidad de NOTAS
3340 IFK%THENDRAW"bm32,120":PRINT#1,"VOLUMEN"
3350 GOSUB4160:IFO=K%THEN3390
3360 A1%=0:A2%=0:DRAW"bm16,160":IFJU%<3THENPRINT#1,"PREMIO:        PUNTAJE:":DRAW
"BM128,176":PRINT#1,"MAXIMO : 00":GOSUB3570:GOTO3380
3370 PRINT#1,"PREMIO1        PUNTAJE1":GOSUB5330:DRAW"BM16,176":PRINT#1,"PREMIO2
    PUNTAJE2":GOSUB5340
3380 DRAW"bm32,136":COLOR1:PRINT#1,"Si comienza, DISPARE-":COLOR15
3390 K%=0:DRAW"bm16,136":PRINT#1,CHR$(210+CO%):ONINTERVAL=10GOSUB3510:INTERVALON
3400 L%=(JU%-1)MOD5:L%=-(L%>0)-(L%>3):ONSTRIGGOSUB3490,3490,3490,3490,3490
3410 ONSTRIGGOSUB3490,3490,3490,3490,3490:IFL%THENSTRIG(L%)ON:STRIG(L%+2)ONELSES
TRIG(0)ON
3420 J%=STICK(L%):IF1<>J%MOD4THEN3420
3430 INTERVALOFF:LINE(16,136)-(23,143),12,BF:FORI%=0TO4:STRIG(I%)OFF:NEXTI%:IFJ%
=1THENGOTO3350ELSEGOTO3290
3440 '
3450 '
3460 '         Subrutinas para
3470 '         interval y strig
3480 '
3490 INTERVALOFF:LINE(16,136)-(23,143),12,BF
3500 STRIG(0)OFF:STRIG(1)OFF:STRIG(3)OFF:NO%=INT(.5*(NC%+NF%)):RETURN4290'Ir a l
a escala
3510 CO%=NOT(CO%):LINE(16,136)-(23,143),12,BF:DRAW"bm16,136":PRINT#1,CHR$(210+CO
%):RETURN
3520 '
3530 '
3540 '    Acualizaci"n de premio
3550 '
3560 '
3570 PM%=(1-.1*NI%)^A1%*(2*(NF%-NC%)+70-5*CT%+9*CN%):LINE(64,160)-(103,167),4,BF
:DRAW"bm64,160":PRINT#1,STR$(PM%):RETURN
3580 '
3590 '
3600 '    Cambiar el intervalo
3610 '
3620 '
3630 ONSTRIGGOSUB3750,3750,3750,3750,3750:ONINTERVAL=10GOSUB3760:DRAW"bm16,72":P
RINT#1,CHR$(210+CO%)
3640 M%=-1:N%=1:PUTSPRITE1,(FNU%(NC%),40),15,1
3650 P%=M%*(NC%-NF%)+NC%:LI%=-M%*(NC%+2):LD%=M%*(NF%-97)+NF%-2
3660 INTERVALON:PUTSPRITE1-M%,(FNU%(P%),40),N%,1-M%
3670 IFL%THENSTRIG(L%)ON:STRIG(L%+2)ONELSESTRIG(0)ON
3680 J%=STICK(L%):IFO=J%MOD2THEN3680
3690 FORI%=0TO50:NEXTI%:IFL%THENSTRIG(L%)STOP:STRIG(L%+2)STOPELSESTRIG(0)STOP
3700 IF1=J%MOD4THEN3780ELSEJ%=.5*(5-J%)
3710 IF((J%=1)AND(M%ANDNF%=LD%ORNOT(M%)ANDNC%=LD%))OR((J%=-1)AND(M%ANDNF%=LI%ORN
OT(M%)ANDNC%=LI%))THENBEEP:GOTO3670
3720 NF%=NF%-M%*J%:NC%=NC%+(1+M%)*J%:IFK%THEN3650
3730 IFJU%<3THENGOSUB3570ELSEGOSUB5330:GOSUB5340
3740 LINE(0,80)-(39,87),12,BF:GOTO3650
3750 PUTSPRITE1-M%,(FNU%(P%),40),15,1-M%:M%=NOT(M%):RETURN3650
3760 CO%=NOT(CO%):LINE(16,72)-(23,79),12,BF:DRAW"bm16,72":PRINT#1,CHR$(210+CO%):
RETURN
3770 N%=1-N%:PUTSPRITE1-M%,(FNU%(P%),40),N%,1-M%:RETURN
3780 INTERVALOFF:PUTSPRITE1-M%,(FNU%(P%),40),15,1-M%:FORI%=0TO4:STRIG(I%)OFF:NEX
TI%
3790 LINE(16,72)-(23,79),12,BF:IFK%THENRETURN3300
3800 IFJ%=1THENRETURN3390ELSERETURN3310
3810 '
3820 '
3830 'Cambiar la cantidad de tentativas
3840 '
3850 '
3860 ONINTERVAL=10GOSUB3950:INTERVALON:GOSUB3950
3870 DRAW"bm128,88":PRINT#1,STRING$(CT%,42):LINE(8*(16+CT%),88)-(8*(17+CT%),95),
12,BF:LINE(0,96)-(119,103),12,BF
3880 J%=STICK(L%):IFO=J%MOD2THEN3880
3890 IFJ%=7ANDCT%=CN%ORJ%=3ANDCT%=15THENBEEP:GOTO3880
3900 IF3>J%MOD4THEN3930ELSECT%=CT%+.5*(5-J%):IFK%THEN3870
3910 IFJU%<3THENGOSUB3570ELSEGOSUB5330:GOSUB5340
3920 GOTO3870
3930 INTERVALOFF:LINE(16,88)-(23,95),12,BF:IFK%THENRETURN3320
3940 IFJ%=1THENRETURN3290ELSERETURN3320
3950 CO%=NOT(CO%):LINE(16,88)-(23,95),12,BF:DRAW"BM16,88":PRINT#1,CHR$(210+CO%):
RETURN
3960 '
3970 '
3980 ' Cambiar la cantidad de notas
3990 '
4000 '
4010 ONINTERVAL=10GOSUB4100:INTERVALON:GOSUB4100
4020 DRAW"bm128,104":PRINT#1,STRING$(CN%,42):LINE(8*(16+CN%),104)-(8*(17+CN%),11
1),12,BF:LINE(0,112)-(119,119),12,BF
4030 J%=STICK(L%):IFO=J%MOD2THEN4030
4040 IFJ%=7ANDCN%=1ORJ%=3ANDCN%=CT%THENBEEP:GOTO4030
4050 IF3>J%MOD4THEN4080ELSECN%=CN%+.5*(5-J%):IFK%THEN4020
4060 IFJU%<3THENGOSUB3570ELSEGOSUB5330:GOSUB5340
4070 GOTO4020
4080 INTERVALOFF:LINE(16,104)-(23,111),12,BF:IFK%THENRETURN3340
4090 IFJ%=1THENRETURN3310ELSERETURN3340
4100 CO%=NOT(CO%):LINE(16,104)-(23,111),12,BF:DRAW"BM16,104":PRINT#1,CHR$(210+CO
%):RETURN
4110 '
4120 '
4130 '     Cambiar el volumen
4140 '
4150 '
4160 ONINTERVAL=10GOSUB4230:INTERVALON:GOSUB4230
4170 DRAW"bm128,120":PRINT#1,STRING$(VS%,42):LINE(8*(16+VS%),120)-(8*(17+VS%),12
7),12,BF:LINE(0,128)-(119,135),12,BF
4180 J%=STICK(L%):IFO=J%MOD2THEN4180
4190 IFJ%=7ANDVS%=1ORJ%=3ANDVS%=15THENBEEP:GOTO4180
4200 IF3=J%MOD4THENVS%=VS%+.5*(5-J%):GOTO4170
4210 INTERVALOFF:LINE(16,120)-(23,127),12,BF:IFK%THENRETURN3360
4220 IFJ%=1THENRETURN3330ELSERETURN3390
4230 CO%=NOT(CO%):LINE(16,120)-(23,127),12,BF:DRAW"BM16,120":PRINT#1,CHR$(210+CO
%):RETURN
4240 '
4250 '
```

```
4260 ' Generaci "n aleatoria de notas
4270 '
4280 '
4290 S%(0)=NC%+INT((NF%-NC%+1)*RND(-TIME)):A1%=-1
4300 FORI%=1TOCN%-1
4310 S%(I%)=NC%+INT((NF%-NC%+1)*RND(-TIME)):IFS%(I%)=S%(I%-1)THEN4310ELSENEXT
4320 IFJU%>2THEN4860
4330 '
4340 '
4350 '        Escala
4360 '
4370 '
4380 CA%=0:CE%=0:NO%=INT(.5*(NC%+NF%))
4390 PLAY"V=VS%;T120L4":FORI%=NC%TONF%:PLAY"N"+STR$(I%):PUTSPRITE0,(FNU%(I%),40),1,0
4400 J%=PLAY(1):IFJ%THEN4400ELSENEXT
4410 '
4420 '
4430 '     Juego de un jugador
4440 '
4450 '
4460 PLAY"V=VS%;T120L4":PUTSPRITE0,(FNU%(NO%),40),1,0:A1%=A1%+1:GOSUB3570
4470 FORI%=CA%TOCN%:IFI%=CN%THENPLAY"R4"ELSEPLAY"N=S%(I%);"
4480 ONSTRIGGOSUB4610,4610,,4610
4490 IFJU%=1THENSTRIG(0)ONELSESTRIG(1)ON:STRIG(3)ON
4500 J%=STICK(JU%-1):IFJ%MOD2THEN4520ELSEJ%=PLAY(1):IFJ%THEN4500
4510 NEXT:GOTO4470
4520 STRIG(0)OFF:STRIG(1)OFF:STRIG(3)OFF
4530 IFJ%=7ANDNO%=NC%ORJ%=3ANDNO%=NF%THEN4490
4540 IF1=J%MOD4THEN4390
4550 NO%=NO%+.5*(5-J%):PUTSPRITE0,(FNU%(NO%),40),1,0:FORJ%=0TO100:NEXTJ%:GOTO4490
4560 '
4570 '
4580 '     Subrutina ON STRIG
4590 '
4600 '
4610 STRIG(0)OFF:STRIG(1)OFF:STRIG(3)OFF:LINE(8*(15+CT%-CA%-CE%),88)-(8*(16+CT%-CA%-CE%),95),12,BF
4620 IFNO%=S%(CA%)THENCA%=CA%+1:LINE(8*(16+CN%-CA%),104)-(8*(17+CN%-CA%),111),12,BF:I%=CN%-1:J%=.5*(VS%+16):PLAY"","L6407V=J%;CDE":GOTO4640
4630 PLAY"","O3S14M40C32":CE%=CE%+1:IFCT%>CA%+CE%THENRETURN4490ELSERETURN4700
4640 IF(CT%=CA%+CE%)OR(CN%=CA%)THENRETURN4700ELSERETURN4510
4650 '
4660 '
4670 '      Final del juego
4680 '       de 1 jugador
4690 '
4700 PP%=INT(.5+PM%*(CA%/CN%)*(1-(CE%/CT%))):LINE(192,160)-(213,167),4,BF:DRAW"bm192,160":PRINT#1,STR$(PP%)
4710 IFCE%>0ORA1>0THEN4730
4720 PLAY"T200V=VS%;L406A207E.D16C+16D06A07D.C+1606B1607C+06A07E.D16C16D06A07D.C+1606B1607C+06AL2BABA4.R2","T200V=VS%;L404A205E.D16C+16D04A05D.C+1604B1605C+04A05E.D16C16D04A05D.C+1604B1605C+04AL2BABA4.R2"
4730 IFMX%<PP%THENMX%=PP%:LINE(192,176)-(213,183),4,BF:DRAW"bm192,176":PRINT#1,STR$(MX%)
4740 K%=0:GOTO3290
4790 '
4800 '
4810 '     Actualizaci "n de premios
4820 '        para dos jugadores
4830 '
4840 P2%=(1-.1*NI%)^A1%*(2*(NF%-NC%)+70-5*CT%+9*CN%):LINE(72,160)-(111,167),4,BF:DRAW"bm72,160":PRINT#1,STR$(P2%)
4850 P4%=(1-.1*NI%)^A2%*(2*(NF%-NC%)+70-5*CT%+9*CN%):LINE(72,176)-(111,183),4,BF:DRAW"bm72,176":PRINT#1,STR$(P4%):RETURN
4860 GOSUB5260
4870 W1%=0:W2%=0:E1%=0:E2%=0:A1%=INT(.5*(JU%-5)):A2%=INT(.5*(4-JU%)):NO%=INT(.5*(NC%+NF%))
4880 '
4890 '
4900 '        Escala
4910 '
4920 '
4930 PLAY"V=VS%;T120L4":FORI%=NC%TONF%:PLAY"N"+STR$(I%):PUTSPRITE0,(FNU%(I%),40),1,0
4940 J%=PLAY(1):IFJ%THEN4940ELSENEXT
4950 A1%=A1%+INT(3-.5*JU%):A2%=A2%+INT(.5*(JU%-3)):PUTSPRITE0,(FNU%(NO%),40),1,0:GOSUB4840
4960 '
4970 '
4980 '  Juego de dos jugadores
4990 '
5000 '
5010 FORI%=W1%+W2%TOCN%:IFI%=CN%THENPLAY"R4"ELSEPLAY"N=S%(I%);"
5020 L%=(JU%<6)*((JU%=5)-1):ONSTRIGGOSUB5150,5150,5150,5150,5150
5030 IFL%=2THENSTRIG(2)ON:STRIG(4)ONELSEIFL%THENSTRIG(1)ON:STRIG(3)ONELSESTRIG(0)ON
5040 J%=STICK(L%):IFJ%MOD2THEN5060ELSEJ%=PLAY(1):IFJ%THEN5040
5050 NEXT:GOTO5010
5060 FORK%=0TO4:STRIG(K%)OFF:NEXT
5070 IFJ%=7ANDNO%=NC%ORJ%=3ANDNO%=NF%THEN5030
5080 IF1=J%MOD4THENIFAY%THEN5030ELSEAY%=NOT(AY%):GOTO4930
5090 NO%=NO%+.5*(5-J%):PUTSPRITE0,(FNU%(NO%),40),1,0:FORJ%=0TO100:NEXTJ%:GOTO5030
5100 '
5110 '
5120 '    Subrutina ON STRIG
5130 '
5140 '
5150 FORK%=0TO4:STRIG(K%)OFF:NEXT:LINE(8*(15+CT%-W1%-W2%-E1%-E2%),88)-(8*(16+CT%-W1%-W2%-E1%-E2%),95),12,BF
5160 IFNO%=S%(W1%+W2%)THENJ%=.5*(VS%+16):PLAY"","L6407V=J%;CDE":W1%=W1%-(JU%<5):W2%=W2%-(JU%>4):LINE(8*(16+CN%-W1%-W2%),104)-(8*(17+CN%-W1%-W2%),111),12,BF:I%=CN%-1:GOTO5180
5170 PLAY"","O3S14M40C32":E1%=E1%-(JU%<5):E2%=E2%-(JU%>4):IFCT%=W1%+W2%+E1%+E2%THENRETURN5410ELSEGOSUB5250:RETURN5020
5180 IF(CT%=W1%+W2%+E1%+E2%)OR(CN%=W1%+W2%)THENRETURN5410ELSEIFAY%THENGOSUB5250
5190 RETURN5050
5200 '
5210 '
5220 '    Subrutinas de cambio e                    impresion del jugador
5230 '
5240 '
5250 JU%=3+(3+JU%)MOD4:PLAY"","L64M25506S1T120V8E"
5260 AY%=0:LINE(72,53)-(79,60),4,BF:DRAW"bm72,53":IFJU%<5THENPRINT#1,"1"ELSEPRINT#1,"2"
5270 RETURN
5280 '
5290 '
5300 '    Subrutina de impresi "n                 actualizada de los premios
5310 '
5320 '
5330 P2%=(1-.1*NI%)^A1%*(2*(NF%-NC%)+70-5*CT%+9*CN%):LINE(72,160)-(111,167),4,BF:DRAW"bm72,160":PRINT#1,STR$(P2%):RETURN
5340 P4%=(1-.1*NI%)^A2%*(2*(NF%-NC%)+70-5*CT%+9*CN%):LINE(72,176)-(111,183),4,BF:DRAW"bm72,176":PRINT#1,STR$(P4%):RETURN
5350 RETURN
5360 '
5370 '
5380 '        Final del juego
5390 '
5400 '
5410 P1%=INT(.5+P2%*(W1%/CN%)*(1-(E1%/CT%))):LINE(192,160)-(213,167),4,BF:DRAW"bm192,160":PRINT#1,STR$(P1%)
5420 P3%=INT(.5+P4%*(W2%/CN%)*(1-(E2KTROFFCT%))):LINE(192,176)-(213,193),4,BF:DRAW"bm192,176":PRINT#1,STR$(P3%)
5430 IF(E1%>0ORA1%>0)ANDJU%<5OR(E2%>0ORA2%>0)ANDJU%>4THEN5460ELSELINE(24,53)-(63,60),4,BF:DRAW"bm24,53":PRINT#1,"BRAVO ";
5440 IFP1%>P3%THENPRINT#1,"1"ELSEPRINT#1,"2"
5450 PLAY"T200V=VS%;L406A207E.D16C+16D06A07D.C+1606B1607C+06A07E.D16C16D06A07D.C+1606B1607C+06AL2BABA4.R2","T200V=VS%;L404A205E.D16C+16D04A05D.C+1604B1605C+04A05E.D16C16D04A05D.C+1604B1605C+04AL2BABA4.R2"
5460 K%=0:GOTO3290
```

EL BASIC MSX Y EL DE PC

Nos referimos a las instrucciones que son iguales o de resultado similar, que permiten usar un programa de MSX en una PC.



Mucho se ha dicho acerca de la compatibilidad de la norma MSX respecto del formato de disco, del CP/M, de los caracteres, de los archivos, etcétera. Estas ventajas permiten acercarse a terrenos más profesionales. Es indudable la satisfacción que se siente al encontrar una máquina que le permita a uno insertar un disco de una PC y poder leerlo. Entonces uno, ya más tranquilo, enciende su computadora y se dispone a hacer un porgrama en BASIC y, una vez concluido se deleita con el "nene" recién nacido. Es en este momento cuando llega un amigo que casualmente necesita un programa que hace las mismas cosas que el nuestro pero para una PC; surge ahí el dilema del BASIC. ¿Será también compatible? ¿Hacemos el programa desde cero en una PC, o usamos el nuestro?

Por eso para clarificar el panorama analizaremos el MSX-BASIC y el BASIC de IBM llamado BASICA pero nos referiremos solamante a las instrucciones que son iguales o de resultado similar:

CONSTANTES

Se definen de igual manera en los dos BASIC: DEFDBL, DEFSNG, DEFINT y DEFSTR.

Los tipos de base que podemos usar

INSTRUCCION	MSX BASIC	BASICA	COMENTARIOS
ABS(n)	SI	SI	
ASC (n$)	SI	SI	
ATN(r)	SI	SI	En radianes
AUTO com, inc	SI	SI	
BEEP	SI	SI	
BLOAD "nom"	SI	SI	
BSAVE "nom" (x)	NO	NO	No son compatibles
CALL rutina (n) *	SI	SI	Teniendo en cuenta rutina
CDBL (n)	SI	SI	
CHR$ (n)	SI	SI	
CINT (n)	SI	SI	
CIRCLE (x, y),r , c, a, b, e	SI	SI	Verificando la resolución
CLEAR var, rntop	SI	SI	
CLOSE n1, n2,...	SI	SI	
CLS	SI	SI	
COLOR t, f, b	SI	SI	En PC modo texto
CONT	SI	SI	
COS (n)	SI	SI	En radianes
CSNG (n)	SI	SI	
CSRLIN (n)	SI	SI	Equivalente en PC a POS (n)
CVD (n$)	SI	SI	
CVI (n$)	SI	SI	
CVS (n$)	SI	SI	
DATA n, m, p, ...	SI	SI	
DEF USR = d.rut	SI	SI	
DELETE c, f	SI	SI	
DIM a (m, n), b (p),...	SI	SI	Equivalente en PC a DIMTABL
EDIT 1	SI	SI	
ELSE	SI	SI	
END	SI	SI	
EOF	SI	SI	
ERASE u, v	SI	SI	
ERL	SI	SI	
ERR	SI	SI	
ERROR	SI	SI	
EXP (n)	SI	SI	
FIELD #n, l1 AS p$	SI	SI	
FILES	SI	SI	
FIX (n)	SI	SI	
FOR TO STEP	SI	SI	
FRE (0)	SI	SI	
FRE (" ")	SI	SI	
GET # c, n	SI	SI	
GOSUB ll	SI	SI	
GO TO ll	SI	SI	
HEX$ (n)	SI	SI	
IF THEN ELSE	SI	SI	
INKEY$	SI	SI	
IMP (p)	SI	SI	En PC se llama INP (P)
INPUT a, b, c,...	SI	SI	
INPUT #c,a,b,c,...	SI	SI	
INPUT$ (n)	SI	SI	
INSTR (cad, arg)	SI	SI	
INT (n)	SI	SI	
KEY ON	SI	SI	
KEY OFF	SI	SI	
KEY n, "mens"	SI	SI	
ON KEY GOSUB ll	SI	SI	
KEY (n) ON, OFF, STOP	SI	SI	
KILL "nom"	SI	SI	
LEFT$(cad, lon)	SI	SI	
LEN (a$)	SI	SI	
LET	SI	SI	Ya no se emplea
LINE (x, y)-(u, v), c, op	SI	SI	
LINE INPUT "mens"; a$	SI	SI	
LINE INPUT # c, a$	SI	SI	

LIST com, fin	SI	SI	También en sus variantes
LLIST com, fin	SI	SI	Idem
LFILES	SI	SI	
LOAD "nom"	SI	SI	Y en su versión, R
LOC (c)	SI	SI	
LOCATE l, c	SI	SI	En PC se agregan comandos
LOF (c)	SI	SI	
LOG (n)	SI	SI	
LPOS (x)	SI	SI	En PC x vale 1, 2 o 3
LPRINT USING	SI	SI	En todos sus formatos
LSET camp$, a$	SI	SI	
MERGE	SI	SI	
MID$ (cad$, pos, lon)	SI	SI	
MKD$ (dbl)	SI	SI	
MKI$ (ent)	SI	SI	
MKS$ (sim)	SI	SI	
MOD	SI	SI	
MOTOR	SI	SI	
NAME "vie" AS "nue"	SI	SI	
NEW	SI	SI	
NEXT a, b,...	SI	SI	
OCT$ (n)	SI	SI	
ON ERROR GOTO (GOSUB)	SI	SI	
OPEN (*)	NO	NO	Difiere la programación
ON STOP GOTO GOSUB	SI	SI	Equivale en PC a ON STRING
ON STRIG GOTO GOSUB	SI	NO	
STOP ON	SI	NO	
ON INTERVAL	SI	NO	
OUT p, dat	SI	SI	
PAINT (x, y), c	SI	SI	En PC se agregan col y lim
PEEK (dir)	SI	SI	
POINT (x, y)	SI	SI	
POKE dir, dat	SI	SI	
POS (0)	SI	SI	
PRESET (x, y)	SI	SI	
PRINT a, b$,...	SI	SI	En todos sus formatos
PRINT # c, a, b,...	SI	SI	
PSET (x, y), c	SI	SI	
PUT #c, n	SI	SI	
READ a, b$, C,...	SI	SI	
REM	SI	SI	En todos sus formatos
RESTORE ll	SI	SI	
RESUME	SI	SI	
RETURN ll	SI	SI	
RIGHT$ (cad$, lon)	SI	SI	
RND (+-n)	SI	SI	
RSET cap$, a$	SI	SI	
RUN ll	SI	SI	
RUN "nom"	SI	SI	
SAVE "NOM"	SI	SI	En todos sus formatos
SCREEN n, (x)	SI	SI	Compatible solo en n
SGN (n)	SI	SI	
SIN (n)	SI	SI	
SPACE$ (n)	SI	SI	
SPC (c)	SI	SI	
SQR (n)	SI	SI	
STEP (x, y)	SI	SI	
STICK (n)	SI	SI	
STOP	SI	SI	
STR$ (n)	SI	SI	
STRIG ON/OFF/STOP	SI	SI	
STRIG (n)	SI	SI	
STRING$ (n, c$)	SI	SI	En todos sus formatos
SWAP n, m	SI	SI	
SYSTEM	SI	SI	En MSX agregar call o _
TAB (col)	SI	SI	
TAN (n)	SI	SI	
TIME	SI	NO	Difiere el formato
TROFF	SI	SI	
TRON	SI	SI	
USR n (x)	SI	SI	
VAL (n$)	SI	SI	
VARPTR (n)	SI	SI	En todos sus formatos
WAIT p, msq1, msq2	SI	SI	
WIDTH col	SI	SI	



son: &H para la hexadecimal, &O para la octal, &B para la binaria (MSX solamente).

VARIABLES

Tienen cuatro variantes: enteras (%), simple precisión (!), doble precisión (#) y cadenas o strings ($).

OPERADORES

Ambos soportan: +, -, *, /, ^.

OPERADORES RELACIONALES

Idem: =, <>, <, >, <=, >=.

OPERADORES LOGICOS

Idem: NOT, AND, OR, XOR, IMP, EQV.

FUNCIONES PERSONALIZADAS

Se mantiene la estructura DEF FN nombre=expresión: FN nom.

No hemos incluido ni DRAW ni SOUND ni PLAY ya que su tratamiento como macro no las hacen totalmente compatibles con PC pero hablaremos de ellas más adelante. La aperturta de canal "OPEN" en BASICA tiene los siguientes formatos: OPEN TIPO$, #CANAL, NOMBRE$, LONG. TIPO$ toma el valor de "I", "O", "R" o "A" según sea lectura secuencial, escritura secuencial, agrega al final o acceso directo; solo para el último se agrega la longitud de registro. Si modificamos esta instrucción en nuestro programa, podremos hacerlo ejecutar sin mayores problemas.

MODO DE ENCARAR LA ENSEÑANZA DEL LOGO

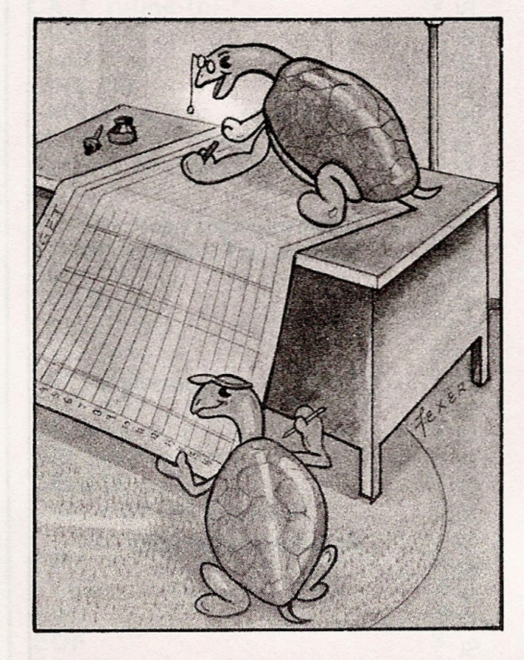
Mucho se habla de este lenguaje y su aplicación en la formación informática del niño. El Licenciado Pablo A.Ger, director del Instituto Nueva Enseñaza de Vicente López clarificó el panorama.

¿Por qué sí LOGO?

LOGO es un lenguaje para programar computadoras, estructurado y simple, que posee un contenido matemático adaptable a los niños.
Es un medio matemáticamente expresivo
Y éste en muchos sentidos (el espacio, el movimiento, las pautas de acción y repetición) no llega a los niños a través de los métodos tradicionales.
Todas estas características hacen que a través de la enseñanza del LOGO, el niño no solamente pueda interactuar con una computadora, sino también aprenda las matemáticas como el conjunto de principios orientadores que rigen el aprendizaje.

¿Por qué no LOGO?

En la enseñanza de un lenguaje de programación de computadoras, cualquiera sea, se enfatiza el aprendizaje de éste y luego el adquirir distintos tipos de conocimientos, pero siempre como algo secundario.
Son dos caminos distintos, si bien se unen en un punto. El que se sigue para enseñar LOGO es mucho más corto y menos trabajoso, pero con una visión de futuro menos amplia y poco duradera.



¿Y entonces qué?

Podemos ver que lo que en definitiva le queda al niño es el conjunto de conductas que pudo aprender a través del estudio del lenguaje y no el LOGO en sí mismo, como un conjunto de órdenes que si bien son fáciles de aprender también son fáciles de olvidar.
El entrenamiento de las mencionadas conductas no excluyen el aprendizaje del lenguaje. Por el contrario a través de él y su práctica, el niño las interioriza, haciéndolas propias, y lo benefician no sólo por haber aprendido un medio matemático expresivo, sino como el medio para descubrir, comprender y expresar su propio proceso de pensamiento.

¿Para qué el LOGO y la computadora?

Es indudable que al introducir una computadora en el aula, se revolucionan los métodos tradicionales de enseñanza.
El maestro ahora adquiere la tarea de diseñar formas ingeniosas de motivar a los niños para que aprenda cosas que naturalmente no aprendía. El docente asume el papel de guía y el alumno, el de creador y dueño absoluto de un conocimiento que le es propio.
Usando como herramientas del aprendizaje una computadora y su lenguaje,

se logra que el niño adquiera una forma de pensar, de razonar, que lo beneficia para poder atrapar sus ideas, expresarlas y comprender los conocimientos que le brindará la escuela, no para colocarlos en un recipiente de ideas, sino encontrándoles un fundamento teórico-práctico.
Todo esto hace que el niño obtenga un aprendizaje natural, el que tiene lugar cuando alguien está en contacto con los materiales adecuados, y que a través de un juego no competitivo llegue a usar la computadora para el desarrollo intelectual.

¿Hasta dónde puede llegar el niño?

A medida que el niño avanza en el conocimiento de su pensamiento y va tomando sus ideas, a través de un entrenamiento que le hace cada vez más fácil comprenderlas, se introduce en un mundo de creatividad absoluta, al ver que sus ideas se transforman en una realidad expresada en la pantalla de la computadora. Aunque no llegue a lograr totalmente lo que pretende, sus errores producen una retroalimentación que le permite aprender sobre ellos.
Para poder avanzar, él mismo va a ir solicitando nuevas herramientas del lenguaje que harán más simple o más sofisticado su trabajo. Esto implica que puede avanzar tanto como su imaginación se lo permita. En la mayoría de los casos, al llegar a cierta edad, el niño solo va dejando su lenguaje de trabajo (en este caso el LOGO) para requerir otro.
Todo lo mencionado lleva a la conclusión de que es el niño el que pone sus límites y no un programa pre-establecido.

¿Se producen cambios en el niño?

Si, se produce un cambio que varía de chico en chico. Existen niños que naturalmente hacen un deporte en forma asombrosa o toman un pincel y dibujan, como artistas. Otros sabrán pensar de una forma innata en ellos, resolverán problemas matemáticos con felicidad o estudiarán con rapidez. A unos les dará la capacidad de reflexión y comprensión, útiles para el aprendizaje, les permitirá la toma de ciertos conocimientos a través de la reflexión conciente, y a aquellos cuya capacidad de reflexión es innata les ayudará a ordenar sus pensamientos con rapidez y eficiencia.

¿Por qué la división por edades y niveles?

A lo largo de la escuela primaria el niño va adquiriendo distintos niveles de captación y abstracción, por ello los grupos de trabajo se dividen por edades y niveles, para lograr de esta manera una homogeneidad de aprendizaje.
Tanto en la escuela primaria como secundaria, los cambios de un año al siguiente son importantes, en cuanto a cantidad y calidad de conocimientos. Cada uno de los cursos, además de estar dividido por edades y niveles de captación, está caracterizado por un logro final a obtener a través de distintas conductas. Como producto final realizará un hito, un proyecto personal de absoluta creatividad donde procurará volcar sus ideas, corregirlas, perfeccionarlas, aprender con sus errores y hasta mostrar sus trabajos, por ejemplo, a sus padres logrando su participación y cooperación.

CONCURSO MENSUAL

Se realizó el primer sorteo de cupones auspiciado por Telemática y los Centros de Asistencia al Usuario. El **primer premio** (una computadora Talent MSX DPC-200) lo ganó **Bárbara Vázquez**, de Florida. El **segundo premio** (un kit educativo cada uno), **Carlos Leeb** de Valentín Alsina, **Laura de Calaveia** de Capital y **Karina Fragoso** de Lanús. El **tercer premio** (una beca cada uno para un curso en el Centro de Asistencia a elección), **Agustín Rosato** de Olivos, **María Concepción Barrionuevo** de Chilavert, **Miguel Holliday** de San Isidro, **Carolina Sarquis** de Capital, y **Nelly Rosa Bruny** de El Palomar.
Este es el primero de una serie de sorteos mensuales. Para todos habrá los mismos premios. Quienes deseen participar deberán dirigirse a cualquiera de las direcciones de los Centros de Asistencia para retirar los cupones y recibir las indicaciones de cómo llenarlos.

21 horas
Sábados de 9 a 13 horas

Ramos Mejía
Bolívar 55 - 1er. piso - Tel. 658-4777
Lunes a Viernes de 9 a 13 y de 14 a 21 horas
Sábados de 9 a 13 horas

San Isidro
Av. Centenario 705 - Tel. 743-9678/747-6094
Lunes a Viernes de 9 a 21 horas
Sábados de 9 a 12 horas

Vicente López
Av. Maipú 625 - Tel. 797-6720
Lunes a Viernes de 10 a 19 horas

INTERIOR DEL PAIS

La Plata - Pcia.de Buenos Aires
Calle 48 No. 529 - Tel. (021) 249905 al 07
Lunes a Viernes de 9 a 21 horas
Sábados de 9 a 13 horas

Bahía Blanca - Pcia.de Buenos Aires
Gral. Paz 257 - Tel. (091) 31582
Lunes a Viernes de 9 a 12 y de 16 a 20 horas

Córdoba - Pcia.de Córdoba
9 de julio 533
Lunes a Viernes de 8 a 12 y de 16 a 20 horas

Villa María - Pcia.de Córdoba
Corrientes 1159 - 2do. piso - Tel (0535) 24311
Lunes a Viernes de 16 a 23 horas
Sábados de 8 a 12 y de 15 a 18

Mendoza - Pcia.de Mendoza
Rivadavia 76 - 1er. piso - Tel. (061) 291348/293151
Lunes a Viernes de 8 a 13 y de 16 a 20 horas
Sábados de 8 a 13 horas

Santa Fe - Pcia.de Santa Fe
Rivadavia 2553 Loc.22 - (042) 41832
Lunes a Viernes de 9 a 12 y de 16 a 19 horas
Sábados de 9 a 12 horas

Rosario - Pcia.de Santa Fe
Barón de Mau 1088
Lunes a Viernes de 8 a 12 y de 15 a 19 horas
Sábados de 9 a 12 horas

S.M.de Tucumán - Pcia.de Tucumán
Bolivar 374 - Tel. (081) 245007
Lunes a Viernes de 9 a 18 horas

SCRASH

AUTOR: JORGE GUILLERMO TENTOR
CLASE: ENTRETENIMIENTO



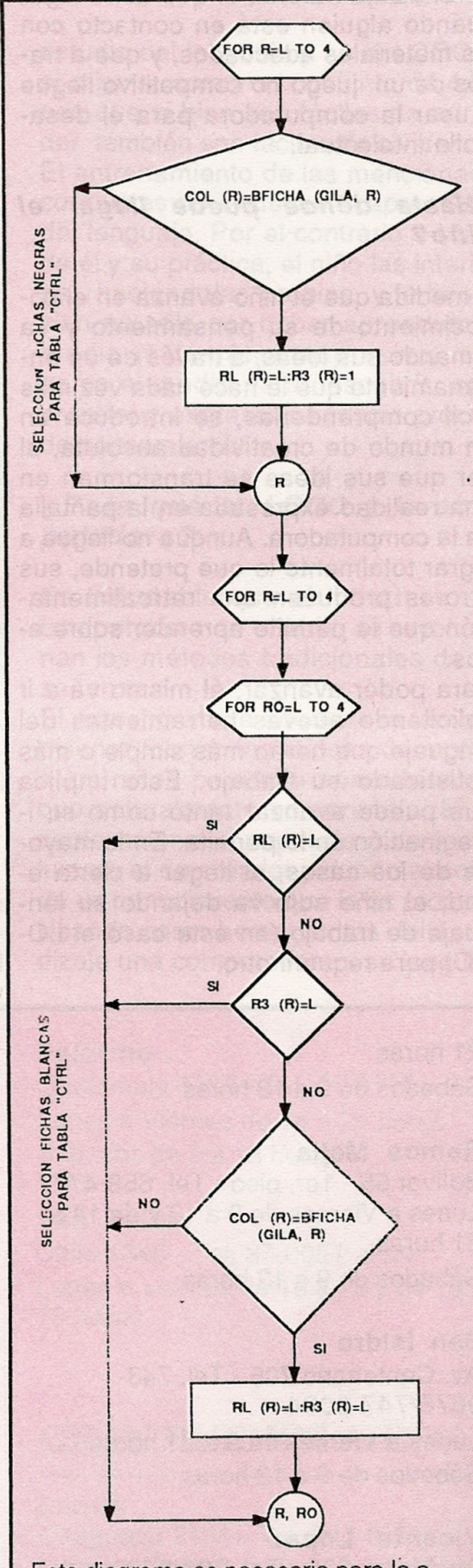
También es llamado juego de los números (se reemplazan los colores por números).

Este juego consiste en adivinar la combinación secreta de los colores elegidos por la computadora. El número de fichas es de cuatro (4) y el de los colores es de cinco (5) (blanco, verde, rojo, amarillo y negro).
Las fichas de colores se eligen mediante las teclas de función. Luego de que el jugador elige las cuatro fichas, la computadora nos indica cuántas coinciden en color y posición, ubicando en la tecla "CTRL" las correspondientes fichas negras. Luego el programa nos indica con las blancas la cantidad de fichas (del jugador) que coinciden en color, pero no en posición con las fichas del código secreto.

DIAGRAMA DE FLUJO DEL BLOQUE DE SELECCION DE FICHAS PARA CTRL

PRESENTACION
GRAFICO
ELECCION 4 COLORES SECRETOS
SONIDO
DEFINE COLORES DEL JUGADOR
BUSQUEDA DE FICHAS COINCIDENTES EN COLOR Y POSICION IMPRESION EN TABLA "CTRL"
4 FICHAS NEGRAS
BUSQUEDA DE FICHAS COINCIDENTES EN COLOR (BLANCAS) IMPRESION EN TABLA "CTRL"
EL NUMERO DE FILA ES=11

DIAGRAMA DE BLOQUES



Este diagrama es necesario para la aclaración de cómo el otro jugador (la computadora) nos indica cuántas fichas negras nos corresponden y cuántas fichas blancas.
Este diagrama corresponde al bloque de líneas 570-790.

Es conveniente destacar que las fichas registradas con puntos negros en la tabla "CTRL" no vuelven a ser registradas para los puntos blancos en la tabla de control ("CTRL"). Se pueden repetir los colores.

Ejemplo:

rojo verde negro rojo ---------
Código secreto
rojo blanco verde amarillo------
Primera fila

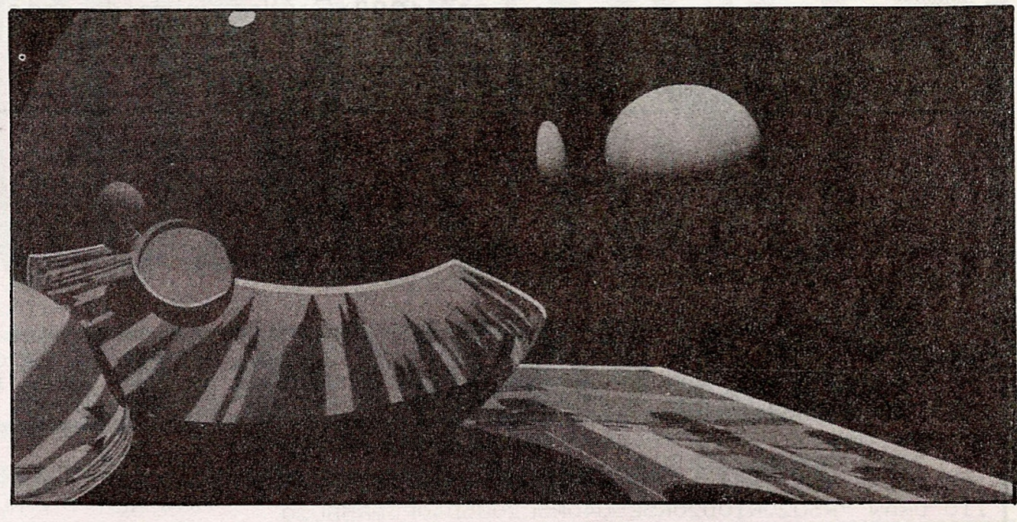
En este caso la tecla "CTRL" nos indicará con un punto negro (rojo) y un punto blanco (verde).

Al seguir jugando pasamos a la segunda fila.

negro verde blanco rojo----------
Segunda fila

En este caso la tabla "CTRL" nos indicará con dos puntos negros (verde-rojo) y un punto blanco (negro).

El juego termina cuando se descubre el código secreto o bien cuando llegamos hasta la última fila (11), en ambos casos se muestra el código secreto y se debe teclear "1" para seguir jugando



VARIABLES IMPORTANTES

FILA: Coordenada "y" para gráficos (fichas).
XCOL: Coordenada "x" para gráficos (fichas).
GILA: Fila en juego.
X: Columna en juego.
KOL: Color elegido por el jugador.
BFICHA (,): Matriz con los colores de las fichas.
ZCOL: Coordenada "x" para fichas de tabla "CTRL".
CTRL: Número de fichas acertadas en color y posición (negras).
COL(): Color de las cuatro fichas secretas.
R1(): Ficha del código secreto.=1 si es coincidente con la ficha del jugador.
R3(): Ficha del jugador.=1 si es coincidente con la ficha del código secreto.

BLOQUES POR NUMEROS DE LINEA

10-110: presentación "SCRASH".
120: Ciclo de retardo.
200-280: Elige 4 colores secretos.
320-450: Impresión fichas del jugador.
460-470: Inicializa variables.
570-630: Registra fichas negras.
700-790: Registra fichas blancas.

800-1300: Gráfico (inicio de cada juego).
870-970: Gráfico (total).
1350-1380: Subrutina sonido.
1420-1430: Subrutina impresiónnivel.

```
10 ON ERROR GOTO 1310
20 COLOR,1,1:SCREEN2:DIM BFICHA(11,4)
30 CIRCLE (125,30),30,6,0,2,1
40 CIRCLE (117,15),13,6,2,3,1
50 PSET(153,33),6
60 DRAW "NE2D2G7D25L20H10L10U5E3U4H2U2R5H5U1E2
   U2H3U1E6U4H4U3H3U15"
70 PAINT(125,30),6
80 FSET (25,145),10
90 DRAW"R20E5U20H5L15U10R20U5H5L20G5D20F5R15D10
   20D5F5BR35R20E5U5L20U30R20U5H5L20G5
   D40F5BR35R5U20R5F9D11R7U14H5E5U20H5L20G5D40F
   5BU30BR5R10U10L10D10"
100 PSET (133,145),10
110 DRAW "R5U20R10D20R5E5U40H5L20G5D40F5BR5BU30
    R10U10L10D10BR30BD30R20E5U20H5L15U10R
    20U5H5L20G5D20F5R15D10L20D5F5BR35R5U20R10D2
    0R5E5U40H5L5D20L10U20L5G5D40F5"
120 FOR R=1 TO 1000:NEXTR
130 SCREEN2:COLOR15,4,1
140 GOSUB 990
150 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
160 REM   ELIGE LOS 4 COLORES
170 REM         SECRETOS
180 REM
190 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
200 FOR S=1 TO 4:RESTORE 240
210 CS=INT(RND(-TIME)*5)+1
220 FOR CA= 1 TO CS:READ COL(S)
230 NEXT CA:NEXTS
240 DATA 1
250 DATA 6
260 DATA 10
270 DATA 12
280 DATA 15
290 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
300 REM     PARTE 2
310 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
320 FILA=10:GILA=0
330 FILA=FILA+15:XCOL=10:GILA=GILA+1
340 FOR X=1 TO 4:XCOL=XCOL+20
350 ON KEY GOSUB 380,390,400,410,420
360 KEY(1)ON:KEY(2)ON:KEY(3)ON:KEY(4)ON:KEY(5)ON
370 GOTO 370
380 KOL=15:RETURN 430
390 KOL=12:RETURN 430
400 KOL=6:RETURN 430
410 KOL=10:RETURN 430
420 KOL=1:RETURN 430
430 CIRCLE(XCOL,FILA),5,KOL:PAINT(XCOL,FILA),KOL
440 BFICHA(GILA,X)=KOL:V=111:GOSUB1350
450 NEXT X
460 ZCOL=120:FOR R=1 TO 4
470 R1(R)=0:R3(R)=0:NEXT:R2=0:R4=0
480 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
490 REM
500 REM          PARTE 3
510 REM
520 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
530 REM SELECCIONA FICHAS NEGRAS
540 REM PARA CONTROL
550 REM
560 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
570 CTRL=0
580 FOR R=1 TO 4
590 IF COL(R)=BFICHA(GILA,R)THEN 600 ELSE 620
600 ZCOL=ZCOL+10:CIRCLE(ZCOL,FILA),3,1:CTRL=
    CTRL+1:V=63:GOSUB1350
610 R1(R)=1:R3(R)=1
620 NEXT R
630 IF CTRL=4 THEN 640  ELSE 700
640 GOSUB1420
650 GOSUB 890
660 GOSUB 990
670 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
680 REM  SELECCIONA BLANCAS
690 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
700 FOR R=1 TO 4
710 FOR RO=1 TO 4
720 IF R1(R)=1 THEN 770 ELSE 730
730 IF R3(RO)=1 THEN 770 ELSE 740
740 IF COL(R)=BFICHA(GILA,RO) THEN 750 ELSE 770
750 ZCOL=ZCOL+10:CIRCLE(ZCOL,FILA),3,15:V=63:
    GOSUB1350
760 R1(R)=1:R3(RO)=1
770 NEXT RO,R
780 GOSUB 1420
790 IF GILA=>11 THEN GOSUB 890 ELSE330
800 GOSUB 990
810 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
820 REM
830 REM   SUBRUTINA GRAFICO
840 REM    Y MUESTRA DE CODIGO
850 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
860 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
870 LINE(103,0)-(119,192),1,BF
880 LINE(171,0)-(255,192),1,BF
890 LINE(21,0)-(101,19),4,BF:KL=10
900 FOR R=1 TO 4:KL=KL+20:CIRCLE(KL,9),5,COL(R)
    :PAINT(KL,9),COL(R):NEXTR
910 OPEN"GRP:"FOR OUTPUT AS#1:COLOR1,1,1
920 PSET(178,180):PRINT#1,"[illegible]"
930 CLOSE:COLOR15,1,1
940 OPEN"GRP:"FOR OUTPUT AS #1
950 PSET(178,180):PRINT#1,"PULSE 1"
960 CLOSE
970 IF INKEY$="1" THEN980ELSE 910
980 RETURN
990 LINE(20,0)-(102,192),5,BF
1000 LINE(32,0)-(87,19),1,BF
1010 OPEN "GRP:" FOR OUTPUT AS #1
1020 PSET(40,1):COLOR15,4,1
1030 PRINT #1,"CODIGO"
1040 PSET(35,11)
1050 PRINT #1,"SECRETO"
1060 LINE(20,0)-(102,192),1,B
1070 LINE(0,0)-(19,192),1,BF
1080 FOR W=25 TO 175 STEP 15
1090 FOR X=30TO90STEP20
1100 CIRCLE(X,W),5,15
1110 NEXT X,W
1120 LINE(120,0)-(170,192),14,BF
1130 PSET(130,1):PRINT#1,"CTRL"
1140 PSET(178,150):PRINT#1,"NIVEL:":CLOSE
1150 FOR W=25 TO 175 STEP 15
1160 LINE(123,0)-(166,9),15,B
1170 DY=20:OPEN "GRP:"FOR OUTPUT AS #1
1180 RESTORE 1240
1190 FOR H=1 TO 5:DY=DY+20:READ HD
1200 PSET(180,DY):PRINT #1,"F";H
1210 CIRCLE(220,DY+3),5,HD:IF H=5 THEN1230
     ELSE 1220
1220 PAINT(220,DY+3),HD
1230 NEXT H
1240 DATA 15,12,6,10,15
1250 COLOR 1,1,1
1260 PSET(178,180):PRINT#1,"[illegible]"
1270 PSET(220,150):PRINT #1,"[illegible]"
1280 CLOSE
1290 COLOR 15,4,1
1300 RETURN 200
1310 RUN
1320 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
1330 REM  SUBRUTINA  SONIDO        @
1340 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
1350 SOUND1,0:SOUND0,V:SOUND7,&B11111110
1360 SOUND8,16:SOUND11,0
1370 SOUND12,1:SOUND13,9
1380 RETURN
1390 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
1400 REM  SUBR.IMPR.NIVEL
1410 REM@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
1420 OPEN"GRP:"FOR OUTPUT AS #1:COLOR1,1,1:PSET
     (220,150)
1430 PRINT#1,"[illegible]":COLOR14,4,1:PSET(220,150):
     PRINT#1,GILA:CLOSE:RETURN
```

3^er CONCURSO

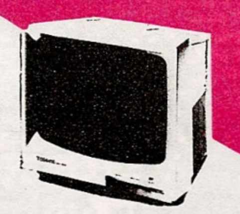
DE PROGRAMAS

auspiciado por TELEMATICA S.A. que proveerá los siguientes Premios:

PRIMER PREMIO

UN PERIFERICO

(a elección entre un monitor, una diskettera y una impresora).



UNA BECA

para trabajar en el Departamento de Investigación y Desarrollo de Telemática S.A.

SEGUNDO PREMIO

UN PERIFERICO

(a elección entre un monitor, una diskettera y una impresora).

ESPECIAL

Entre los programas recibidos, algunos de ellos podrán ser editados por SYSTEMAC S.A., reconociéndose los derechos de autor

En caso de que el ganador no pueda utilizar la beca,será ofrecida a quien obtenga el segundo premio, y si éste tampoco pudiera aprovecharla se otorgará a alguno de los participantes del certamen que se hubiera destacado.

Se premiará el mejor software de cualquier clase (juegos, utilitarios, científico o comercial).

BASES: No sólo será indispensable que el programa enviado en caset ó disket funcione correctamente, sino que además debe cumplir con ciertas reglas:

- Programación estructurada en bloques fácilmente diferenciables.
- Fácil seguimiento del mismo y detalle de éste como parte de su documentación. (Diagrama de bloques con los números de línea que los identifiquen)
- Aclaración y clara explicación de los algoritmos utilizados, deben figurar como parte de la documentación.
- Las variables y/o direcciones de memoria utilizados también se deben incluir en esta documentación.
- Listado de nemónicos assembler y la localización en memoria si es que se utiliza este tipo de lenguaje.
- Calidad y originalidad de gráficos, sonidos y pantallas de menú.

Los trabajos deberán enviarse antes del 30 de noviembre próximo (cierre del certámen) a: Paraná 720, piso 5º, (1017) Capital Federal.

CERVEUX

¿C EL LENGUAJE DEL FUTURO?

El lenguaje C está adquiriendo un auge muy grande en los círculos computacionales. Su velocidad y flexibilidad lo hacen ideal para la construcción de programas comerciales, sistemas operativos y hasta de programas hogareños.

Erase una vez... La historia comienza, como muchas otras que tienen que ver con el desarrollo de la computación, en los laboratorios Bell de la A T & T. Allí, ante el requerimiento de un sistema operativo poderoso para máquinas chicas, con capacidades similares a las de los grandes operativos de los Mainframes, Ken Thomson empieza a programar sobre un ensamblador hasta llegar a una primera versión de UNIX. Pero las necesidades son mayores aún. La tarea se complica tanto que Thomson decide utilizar un lenguaje de alto nivel (el nivel de un lenguaje tiene que ver con su grado de abstracción, es decir, con cuán fácil resulta hacer un programa; por ejemplo el más bajo nivel es el código máquina mientras que de alto nivel son BASIC o Pascal y uno de los más altos es el NATURAL de IBM, un sistema inteligente de Bases de Datos).

A la hora de elegir, prefiere desarrollar un lenguaje propio (cosa bastante común para esta gente), y crea una versión de un compilado de BCPL, un poco difundido hijo del Pascal de Niklaus Wirth. Abreviando lo bautiza B.

Pero hace falta más, y Thomson está en una encrucijada: trabajar sobre un ensamblador significa largas horas de codificación y se hace inmensamente tedioso y complicado; por otro lado, el B le facilita la programación pero baja considerablemente el rendimiento en velocidad y, sobre todo, genera objetos muy grandes.

Extenuado sobre la DEC-PDP11 (máquina clave también en la historia de la programación moderna), Ken pasa el problema a alguien de confianza: Dennis Ritchie.

Este señor tiene eso que los programadores suelen llamar "toque maestro", y lo que hace es empezar de nuevo a codificar en B. Pero mientras desarrolla el UNIX va modificando paralelamente el B hasta convertirlo en otro lenguaje al que llama, obviamente, C . Y aquí es donde Ritchie cambia la historia, prosiguiendo las enseñanzas y líneas de razonamiento de los grandes maestros: Dikstra, Knutt y Wirth, padres de la programacion.



Parte de las rutinas mínimas en ensamblador hasta llegar a un protocompilador de C. A partir de ahí, el verdadero compilador, ¡está hecho en C! Mentes recursivas, si las hay...

EL COMPILADOR DE C

Una vez finalizado el Sistema Operativo UNIX, el C quedó definitivamente vinculado a éste. De manera que el UNIX es el entorno ideal para desarrollar software en C y C es el lenguaje ideal para cualquier implementación que deba ejecutarse sobre este S.O. Con el correr del tiempo, la potencia del lenguaje y sus singulares características le han hecho ganar un lugar privilegiado ante los ojos de los programadores.

Aunque es un lenguaje más de propósitos generales, el C ganó aceptación rápidamente a partir de su independencia del hardware. ¿Qué quiero decir? Que el C trabaja sobre objetos sencillos y comunes a la mayoría de las computadoras. Originalmente, sus tipos de datos son solamente caracteres (de un byte), números y direcciones de memoria, y sus operaciones aritmético-lógicas son las mínimas existentes en cualquier máquina actual. No tiene estructuras como strings, sets o listas. Tampoco tiene implementadas funciones de Entrada/salida.

Eso hace que un compilador de C pueda ser implementado sobre cualquier computador. ¿Entonces, dónde está su potencia? En sus estructuras de control: bucles y condicionales (FOR, WHILE, IF) son en el C tan funcionales que la programación se hace sumamente dinámica; y por supuesto, las aparentes limitaciones tienen su porqué. Todas las funciones que no se hallan implementadas en el compilador mismo están incorporadas en las llamadas bibliotecas del C.

De esta manera se pueden utilizar todos los tipos de estructuras de datos

o funciones de entrada y salida que antes mencionábamos mediante llamadas a dichas bibliotecas. La gracia es precisamente que son llamadas solamente las necesarias para cada programa, lo que hace que el programa objeto no contenga rutinas que no se utilizan. Esto obviamente optimiza mucho el largo del código.

El C es, por ser tan conciso, bastante fácil de aprender, y algunos de sus recursos específicos hacen la delicia de los aficionados a la programación. Un ejemplo concreto es que cualquier rutina desarrollada por el usuario puede ser definida como una función más de la biblioteca, siguiendo así la tendencia moderna de los lenguajes extensibles. Y cuando el problema se pone difícil, o los tiempos de ejecución deben ser reducidos, existe la posibilidad de definir una biblioteca con rutinas en código máquina.

Si bien el C está hecho para correr originariamente bajo UNIX, esto no invalida su utilización en otros sistemas. Una de sus características principales es precisamente la transportabilidad. Utilizando las bibliotecas estándar el mismo programa fuente puede ser compilado y ejecutado en distintas computadoras. Y aun cuando fuera necesario incluir partes en código máquina, éstas, definidas como bibliotecas en el compilador original, pueden ser reescritas en el compilador de la máquina destino, trabajo que no puede compararse a tener que reescribir todo el programa.

APLICACIONES DEL C

La velocidad que alcanza y lo compacto que resulta el código generado hizo que muchas de las aplicaciones cuya resolución normalmente se plantearía en Assembler fueran resueltas utilizando C.

Es muy normal que el programador genere algunas bibliotecas en Assembler para controlar el hardware sobre el que va a correr el programa, pero una vez definidas, éstas quedan incorporadas al compilador. Generando las mismas funciones de biblioteca para distintos entornos del programa que las utiliza puede ser recompilado en todos ellos, aumentando muchísimo la cantidad de usuarios potenciales.

Mediante estas bibliotecas uno va personalizando su compilador. Por ejemplo, podríamos crear una librería para gráficos (para poner un punto, una línea, un círculo, definiríamos funciones: PUNTO (X,Y), LINEA (X1, Y1, X2, Y2), CIRCULO (X, Y, R) u otras) o podríamos hacer una biblioteca de funciones para manejar archivos con estructura en árbol. De allí en más, cada vez que en un programa necesite hacer una operación de este tipo sólo debo avisar en qué fuente debe ser incluida tal o cual biblioteca, y luego utilizo mis funciones como cualquiera de las originales. Este método de trabajo es realmente cómodo y eficiente, lo que se comprueba al saber que gran parte del software famoso está escrito en C. Por ejemplo, el MS-DOS desde su versión 3.0 en adelante, el DBase III (y el III Plus), el LOTUS 1-2-3, y muchísimos más. Incluso la mayoría de los compiladores de otros lenguajes (Pascal, BASIC, Prolog, LISP, ensambladores), y por supuesto, los de C, están escritos en C.



El MSX-C

Para adentrarnos un poco en el lenguaje mismo vamos a hablar del compilador de C que más nos interesa: el MSX. Este es un compilador igual a cualquier otro, con una biblioteca estándar, salvo por una limitación: entre las variables numéricas no existen los tipos FLOAT (punto flotante), ni los LONG INTEGER (enteros grandes, normalmente de 4 bytes). El único tipo numérico es el INTEGER, de 2 bytes de largo (puede contener números entre -32768 y 32767 o, declarándolo como UNSIGNED INTEGER, entre 0 y 65535). El núcleo del compilado contiene:

- Los operadores aritméticos "+", "-", "*", "/" y "%" (resto de la división), que admiten todo tipo de combinaciones, agrupados con paréntesis.
- Los relacionales ">", "<", "=", ">=", "<=", "==" (igual que), "!=" (distinto de).
- Los de manejo de bits "&" (and), "!" (or), "^" (xor), "<<" (rotación a la izquierda) y ">>" (rotación a la derecha).
- Y los maravillosos operadores de incremento "++", y decremento "--".

Estos operadores pueden usarse sobre una variable dentro de cualquier expresión; por ejemplo: "A=B++" carga en A el contenido de B y luego incrementa B en 1. Pero "A=++B" incrementa primero B y luego carga el contenido en A. Todo depende de la posición en que coloque el doble signo. Exactamente lo mismo puede hacerse con "--", que decrementa en 1 el contenido de la variable.

Las estructuras de control son similares a las de Pascal: WHILE, IF, FOR, DO WHILE (que , como el REPEAT del Pascal, evalúa la condición al final del ciclo), y existe una sentencia SWITCH que trabaja en forma similar al CASE.

Las instrucciones que deben ejecu-

CERVEUX

tarse según la condición evaluada en el bucle van encerradas entre llaves, aun cuando no estén en la misma línea, y cada una de ellas debe ser finalizada por ";" como en Pascal.

Cada una de las rutinas es definida como una función. Se llaman así porque pueden emitir un resultado y se le pueden pasar parámetros. Por ejemplo:

```
int potencia ( A , B )
int  A , B;
     (     int  P , Q;
           P = A;
           for ( Q = 0 ; Q != B ; Q++
                 P * = A;
           return ( P );
```

Ésta función eleva a la potencia B un número A. Al definirla, "int" especifica el tipo INTEGER para el resultado de la función. Al empezar, primero declaramos el tipo de los parámetros (en este caso también INTEGER), abrimos llaves y declaramos las demás variables que vamos a utilizar. Cargamos en P el valor de A y comenzamos el bucle FOR, donde, entre paréntesis y separadas por ";", van tres expresiones.

La primera es la inicialización del lazo; en este caso ponemos Q en 0.

La segunda es la condición que es evaluada antes de reiniciar un ciclo. Aquí preguntamos si Q es distinto a B.

Y la tercera expresión indica lo que debe hacerse al iniciar cada ciclo. En el ejemplo, incrementar Q.

Dentro del lazo vamos acumulando en P su propio contenido por A. La sintaxis de esta asignación es característica del C y es lo mismo que "P = P * A", que también sería válido. La idea es entonces que multiplicamos A por A, B veces. En este caso, como hay una sola instrucción dentro del bucle, no necesitamos encerrarla entre llaves.

Una vez que salimos del bucle, ya podemos salir de la función, devolviendo el resultado que nos quedó en P; de eso se encarga la instrucción return(), donde entre paréntesis indicamos la variable que pasamos como salida de la función.

Si quisiéramos desde otra función guardar en una variable cualquiera (por ej: N), 14 elevado a la quinta, sólo tenemos que hacer:

N = potencia (14 , 5);

Las variables declaradas en una función pertenecen solamente a ésta y son independientes de las declaradas al principio del programa que son comunes a todas.

Una función también puede llamarse a si misma, ya que el C admite recursividad. Pero como esto genera mucho espacio de reserva para los stacks de variables, podemos avisarle al compilador si no vamos a utilizar recursividad, y esos espacios no son generados. Este tipo de advertencias al compilador se indican con la sentencia PRAGMA: PRAGMA NONREC indica si no hay recursividad, SPACE OPT da prioridad al ahorro de espacio, aunque el código generado sea más lento, y SPEED OPT da prioridad a la velocidad de ejecución, aunque el objeto quede más largo.

LA BIBLIOTECA ESTANDAR Y LA COMPILACION

En esta biblioteca provista con el MSX-C, tenemos todas las instrucciones de Entrada y Salida, manejo de



strings, etcétera, que no contiene el compilador en sí mismo . La llamamos estándar porque contiene lo mismo que cualquier otro compilador de cualquier otra máquina, lo que hace posible la tan mencionada transportabilidad.

Las bibliotecas a incluirse son determinadas en el programa mediante la sentencia #INCLUDE, y una vez que el fuente está listo, se graba y se compila ejecutando desde el DOS el comando "C", que llama al "C.COM", el compilador.

Este compilador (la versión 2.0) genera un programa fuente en Assembler de 8080, un microprocesador anterior al Z-80, pero con el mismo código de máquina excepto por algunas instrucciones "extras" del Z-80. De todos modos tenemos noticias de que pronto estará la versión 3.0 que genera código Assembler de Z-80.

Lo cierto es que este programa Assembler debe ser ensamblado con el Macro-Assembler 80 (M80) y "linkeado" con el Linker 80 (L80), para obtener el programa objeto que tendrá el mismo nombre del fuente pero con extensión ".COM", lo que permite ejecutarlo desde el Sistema Operativo.

Y ESTA HISTORIA CONTINUARA...

¿Qué puede pasar con el C?. Indudablemente este lenguaje tiene caratcterísticas muy interesantes que facilitan la labor del programador acostumbrado a trabajar con lenguajes estructurados. Además, su velocidad, levemente menor al Assembler, y su transportabilidad lo están llevando a desplazar al Assembler en la elaboración de sistemas operativos, compiladores y otras aplicaciones. La razón es muy sencilla: el tiempo de investigación que lleva el descubrir los secretos de un microprocesador no se justifica si en poco tiempo se debe comenzar el estudio de uno nuevo. Con el C, en cambio, los ajustes que hay que hacer son mínimos, justamente por la mencionada transportabilidad.

En cuanto al usuario común, el C no es difícil de aprender y representa una buena opción para el que siente que el BASIC le está "quedando chico" y no quiere introducirse en el mundo del Assembler.

INGRESO DE TEXTO

Con esta rutina tratamos de solucionar un problema bastante común: el largo de las strings que deben ser ingresadas en un programa.

Esto es casi imprescindible si usan archivos relativos, ya que el usuario sabrá que todo lo que logre ingresar queda grabado.

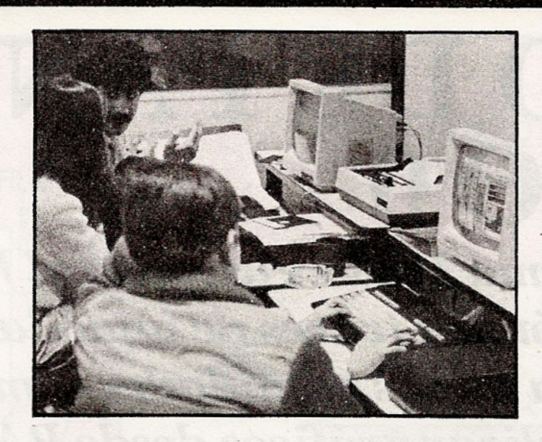
El uso de la rutina es muy sencillo y

```
10 CLS
20 LOCATE 2,10:PRINT"Nombre:
30 QX=10:QY=10:QL=20:GOSUB 10000
40 PRINT"*";QA$;"*"
99 END
99993 '--------------------------
99994 '-- Ingreso Controlado --
99995 '--------------------------
99996 '-Martín Salías-----------
99997 '--------------------------
99998 '-Club de Usuarios MSX---
99999 '--------------------------
10000 QA$="":QP=0
10010 LOCATE QX+QP,QY,1
10020 QJ$=INPUT$(1)
10030 IF ASC(QJ$)=13 THEN PRINT:
      RETURN
10040 IF ASC(QJ$)=8 AND QP>0 THEN
   QP=QP-1:QA$=LEFT$(QA$,QP):GOTO
10080
10060 IF QP=QL THEN BEEP:GOTO 10010
10070 QP=QP+1:QA$=QA$+QJ$
10080 LOCATE QX,QY:PRINTQA$;" "
10090 GOTO 10010
```

todas las variables empiezan con la letra "Q", para evitar conflictos con el programa donde la utilice.
Tienen que pasar los parámetros QX y QY, que indican la posición a partir de la cual debe aceptarse el texto, y QL donde se especifica el largo máximo de éste. Al llamar a la rutina 10000, aparecerá el cursor como si fuera un INPUT común y corriente, salvo que al llegar al máximo hace un BEEP y no permite que continúe ingresando. Si algo me queda cortado, puedo volver hacia atrás con la tecla BS y corregirlo.
Les dejo como propuesta que expandan esta rutina agregándole, por ejemplo, el movimiento de los cursores, la validación del ingreso (si son números o letras) y un montón de cosas más que pueden ocurrírseles al usarla.

CURSO DE DIAGRAMACION ESTRUCTURADA

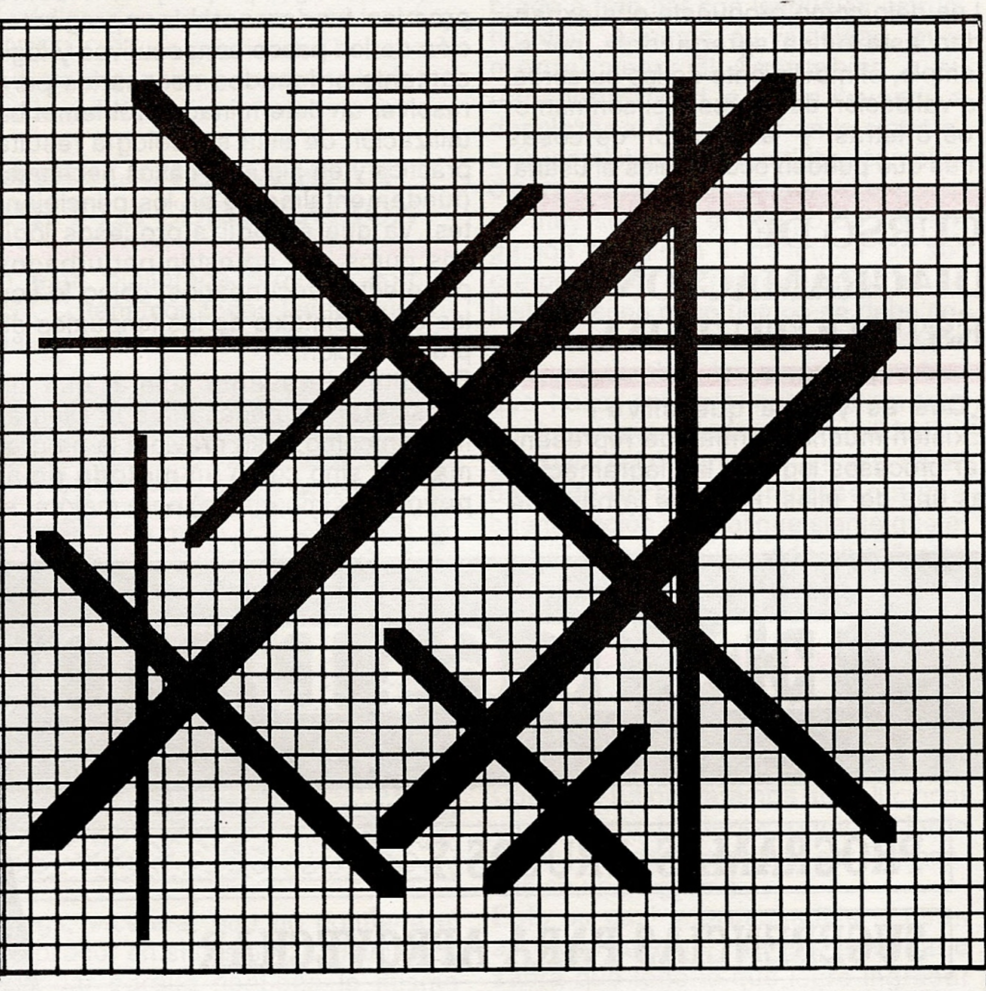
¿Qué es y para qué sirve?
Existen muchas formas de representar procesos lógicos; la diagramación es una de ellas. Parte de la base de



una simbología propia y tiene como premisa fundamental la representación de los pasos consecutivos y lógicamente ordenados necesarios para resolver un determinado problema. La utilización de esta simbología resulta práctica y en algunos casos necesaria (fundamentalmente en los principiantes) ya que simboliza procesos lógicos puros que no están perturbados por efectos secundarios, como lo son los de la sintaxis de los lenguajes de programación.
Resulta interesante aclarar que no consideramos necesaria a la Diagramación como paso previo a la programación, sino como un **método de apoyo** (muy importante) para mejorar el desarrollo de procesos lógicos.
Dentro de la Diagramación existen diversas simbologías, pero todas están enmarcadas dentro de dos grandes grupos: lineales o estructuradas.
En nuestro caso adoptamos el segundo tipo debido a su mayor claridad y eficiencia en la representación de un problema.
Nuestro curso comienza profundizando estas ideas. Se explican el concepto de diagrama de lógica y las estructuras básicas de control: **secuencia**, **decisión** e **iteración**.
También el concepto de variables y sus diferentes tipos, las operaciones aritméticas y lógicas, qué es una bandera, etcétera.
Luego comienzan a verse las estructuras de datos: archivos, tablas, acumuladores , contadores. Con estos elementos se comienza a diagramar utilizando decisiones y rutinas que se aplican a cortes de control, actualización de archivos, manejo de tablas y matrices, y manejo de pantalla.
Todo esto es **universal** en cualquier lenguaje de programación que elijamos para trabajar, y conociendo la forma de resolver problemas lógicos, podremos dominar mucho más fácilmente cualquiera de ellos.

MANEJO DE PANTALLA EN MODO 0

Para muchos, la norma MSX es la natural heredera de la TI-99/4A. Explicamos cómo a través de aquélla se han rescatado y mejorado la mayoría de sus cualidades como por ejemplo las facilidades para realizar gráficos desde BASIC en modo 2 .



Es indudable que la gran mayoría de usuarios que hoy dia transitan los senderos de la microcomputacion, y que tienen más de 3 años de experiencia en el tema, han dado sus primeros pasos de la mano de la TEXAS TI 99/4A.

Quienes han seguido la evolución más lógica, han continuado el camino con la norma MSX, que vendría a ser su natural heredera. Dado que de la TI99 se ha tomado el chip de video (TMS 9919/29) el manejo de pantalla que poseen las microcomputadoras MSX es muy similar al de la Texas, y el modo 1 (SCREEN 1) que permite trabajar en 32 columnas y con SPRITES, es un fiel reflejo de sus características.

En las micro MSX se han rescatado la mayoría de las cualidades que la TI99 ofrecía sobre los demás equipos de su época, y así se hicieron muchas mejoras en ciertos casos con respecto al sistema original. Como ejemplos se pueden mencionar las facilidades para la realización de GRAFICOS desde BASIC en el modo 2 (SCREEN 2) de ALTA RESOLUCION, o la facilidad para cambiar de modos tanto desde los programas en BASIC, con el comando SCREEN ..., como desde el teclado mediante el uso de las teclas de funcion que pueden ser reprogramadas (1).

Pero en todo cambio se pierden algunas cualidades, y más cuando dicho cambio se produce de la mano de otra empresa, en este caso la MICROSOFT Corp. quien adaptó su M-BASIC a las MSX, sin reparar demasiado en el excelente EXTENDED BASIC que traía la TI99

EL MANEJO DE PANTALLA EN LA TI99

En el camino quedaron pues ciertos comandos que posibilitaban un excelente manejo de pantalla. Por ejemplo el DISPLAY AT, que permitía no solo ubicar con precisión el cursor para escribir, sino además agregar un BEEP , blanquear ciertas posiciones de la pantalla o limpiarla por completo; el ACCEPT AT, mucho mas versátil que el INPUT, permitía aceptar el ingreso de datos desde cualquier lugar de la pantalla, emitir un BEEP cuando la máquina estaba lista para el ingreso , blanquear ciertas posiciones de la pantalla o limpiarla por completo, limitar la cantidad de caracteres y el tipo de los mismos (numéricos, alfanuméricos, dígitos y especificar ciertos caracteres), y otros más de no tanta importancia.

Pero con un poco de imaginación, las funciones que mencionamos previa-

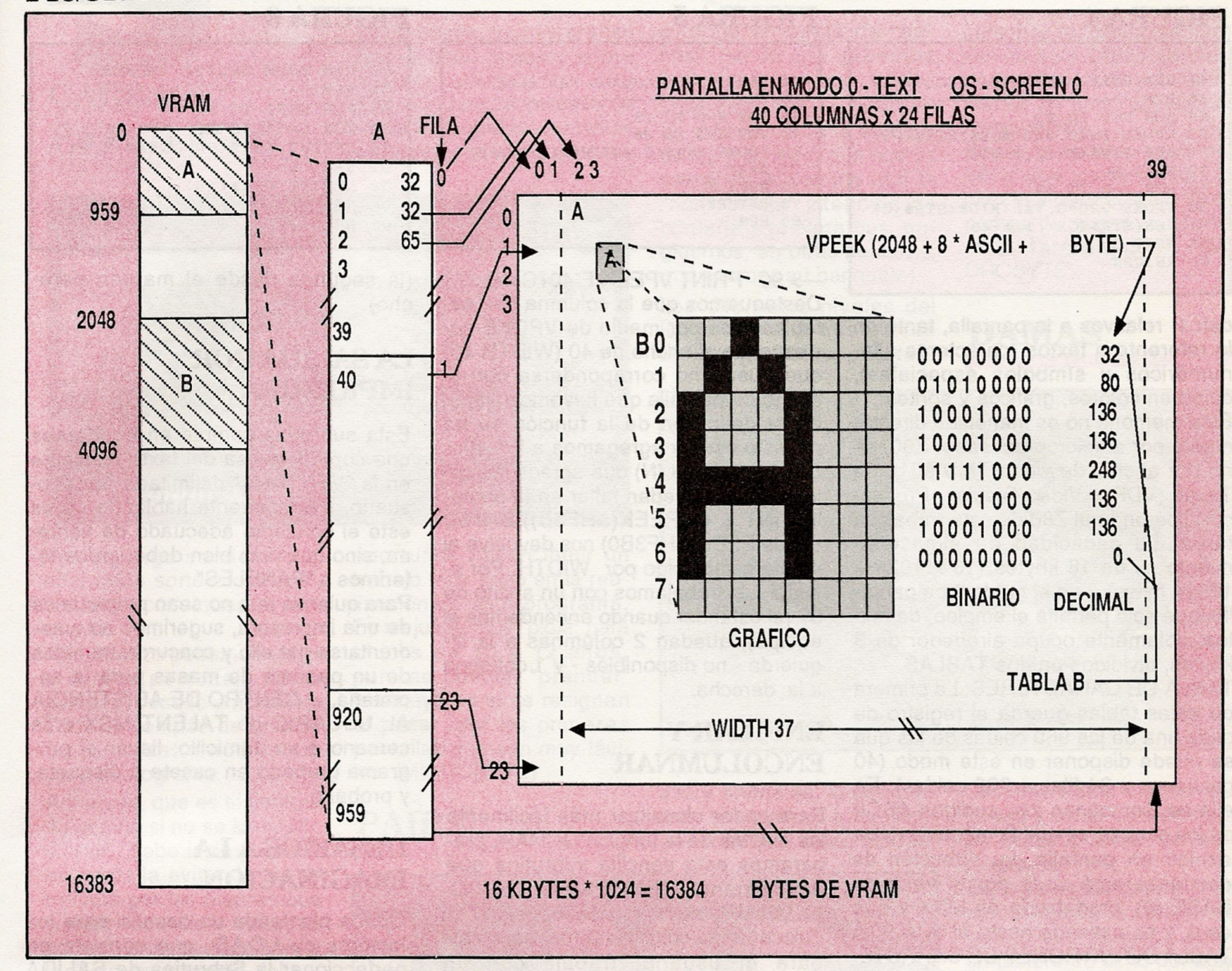
FIGURA 1

```
10 REM MANEJO PANTALLA SCREEN 0
12 REM ------------------------
14 REM funcion call gchar TI99
16 REM
18 REM  por Gustavo O. Delfino
20 REM
22 DEF FN GCHAR(C,F)=VPEEK(F*40+
 C+INT((40-PEEK(&HF3B0))/2+0.5))
24 SCREEN 0:WIDTH 37:COLOR 1,3
26 GOSUB 48
28 GOSUB 68
30 LOCATE 0,17:PRINT "FILA:
 Desde ";:INPUT FI
32 LOCATE 21,17:PRINT "Hasta ";:
 INPUT FF
34 LOCATE 0,18:PRINT "COLUMNAS:
 Desde ";:INPUT CI
36 LOCATE 21,18:PRINT "Hasta ";:
 INPUT CF
38 GOSUB 86
40 END
```

FIGURA 2

```
DEF FN GCHAR(C,F)=VPEEK(F*40+C+
INT(40-PEEK(&HF3B0))/2+0.5))
```

FIGURA 3



mente pueden ser reemplazadas por otras del MSX-BASIC. Por ejemplo, para ubicar un mensaje en cierto lugar de la pantalla, recurrimos a LOCATE C,F:BEEP:PRINT "mensaje...". La que resulta muy difícil de reemplazar es la función GCHAR.

LA FUNCION GCHAR DE LA TI99

En realidad el GCHAR, (GET CHARacter) no era estrictamente una función preprogramada, sino un SUBPROGRAMA predefinido, que debía ser llamado con CALL. Este leía un carácter ubicado en cualquier lugar de la pantalla, y devolvía en una variable numérica el código ASCII del mismo.
Esta herramienta sumamente poderosa permitía un excelente manejo de la pantalla y, entre otros usos que hemos mencionado de ella, uno de ellos era el de permitir armar rutinas de vuelco de pantalla por impresora, seleccionando inclusive partes de la misma a modo de VENTANAS.
Hemos preparado un pequeño programa de demostración de uso de esta poderosa función para la línea MSX en la figura 1.

LA DEFINICION DE FUNCIONES MSX

Ciertas tareas que deben ser empleadas en varias ocasiones durante el transcurso de un programa pueden ser definidas previamente y llamadas por su nombre cuando se requiera su uso. El comando específico para definirlas es DEF y la sintaxis: DEF FN NOMBRE (v1, v2, ...vn) que es equivalente a DEFinir FunciόN NOMBRE (v1, v2, ...vn).
Es usual ver dicha sintaxis "alterada" por los usos prehistóricos de las épocas en que las computadoras traían muy poca memoria disponible para el usuario, y por consiguiente ¡había que ahorrarla...!. Se recurría en esos casos a eliminar los espacios entre una y otra palabra o comando, y a otros artilugios que, si bien lograban su cometido, lo hacían a costa de perder la legibilidad del conjunto y dificultar notablemente la tarea de DEPURACION de los programas. Desgraciadamente hoy día, ya superadas las limitaciones de memoria disponible (¡y más aún en la norma MSX!) persisten estas costumbres a modo de vicios de difícil erradicación. De la forma DEF FN NOMBRE (V1, V2, ... VN) suele verse DEFFNN (V1, V2,... VN) que aconsejamos no utilizar.
La funcion GCHAR queda definida en la figura 2.
Para poder explicar la estructura de esta función debemos recurrir a tratar, aunque sea muy sucintamente, el manejo de la VRAM de la norma MSX.
Esta denominación (VRAM) es la más usada en la bibliografía disponible, pero suele llamársela más corrientemente: MEMORIA DE VIDEO.

LA MEMORIA DE VIDEO MSX

Las microcomputadoras que han adoptado la norma MSX en el mundo poseen una memoria RAM especialmente diseñada para almacenar los

FIGURA 4

```
44 REM SUBR. ENFILAR+ENCOLUMNAR
46 REM -------------------------
48 FOR F=0 TO 23
50 VPOKE F*40,VAL(MID$(STR$(F),
   LEN(STR$(F)),1))+48
52 NEXT F
54 FOR C=0 TO 39
56 VPOKE 920+C,VAL(MID$(STR$(C)
   LEN(STR$(C)),1))+48
58 NEXT C
60 RETURN
```

datos relativos a la pantalla, tanto en lo referente a textos (caracteres alfanuméricos y símbolos especiales), como en colores, gráficos y sprites.
Esta memoria no es manejada directamente por el microprocesador Z80, sino por el chip de video TMS 9919 de Texas (VDP = Video Display Processor) liberando al Z80 de esta trabajosa tarea. La capacidad de almacenamiento es de 16 kbytes (16 x 1024 = 16384 bytes) y en el modo 0 de pantalla, que solo permite el empleo de textos, solamente ocupa alrededor de 3 kbytes, divididos en dos TABLAS.
TABLA DE CARACTERES: La primera de estas tablas guarda el registro de cada una de las 960 celdas de las que se puede disponer en este modo (40 columnas x 24 filas = 960 celdas). En ella se almacenan los codigos ASCII de los caracteres que la máquina debe exhibir en pantalla. La dirección de comienzo está dada por el valor de BASE (0), predefinida en MSX y que es 0, y se extiende hasta el byte 959.
TABLA DE PATRONES DE CARACTERES: La segunda de las TABLAS del modo 0 es la correspondiente a los patrones de los 256 caracteres disponibles (ver Figura 3) cada uno de los cuales está definido por ocho bytes (del 0 al 7). La dirección de comienzo de esta tabla está dada por BASE(2) y es 2048, y se extiende hasta el byte 4096. Longitud 4096 - 2048 = 2048 bytes , correspondientes a 256 caracteres x 8 bytes c/u. Para poder escribir en esta memoria VRAM (VIDEO RAM), se utiliza el comando VPOKE cuya sintaxis es: VPOKE (dirección, código ASCII), y para leer empleamos el VPEEK, comando cuya sintaxis es: VPEEK (dirección).
Si por ejemplo queremos colocar una Z (código ASCII=90) en el casillero correspondiente a la fila tercera (0-1-2...) y a la columna cuarta(0-1-2-3...), ingresaremos:
VPOKE (2*40+3,90) ---> Z VPOKE (F*40+C,ASCII)
Por el contrario, si deseamos leer el código del carácter que ocupa un determinado lugar en la pantalla, por ejemplo fila tercera, columna cuarta, tipearemos: PRINT VPEEK(2*40+3+2)

FIGURA 5

```
64 REM SUBR. LLENADO PANTALLA
66 REM ----------------------
68 FOR F=0 TO 15
70 FOR C=2 TO 38
72 VPOKE F*40+C,INT(RND(1)*26+65)
74 NEXT C
76 NEXT F
78 RETURN
80 REM - - - - - - - - - - - - -
```

---> 90 PRINT VPEEK(F*40+C+N)
Destaquemos que la columna que establecemos por medio de VPOKE corresponde al ancho de 40 (WIDTH 40) que puede no correponderse con el ancho de pantalla que hayamos fijado. En la definición de la función se ha previsto esto, y agregamos a F * 40 + C, un sumando (N) que agrega las columnas que puedan faltar en la pantalla: INT (40-PEEK(&HF3B))/2+0.5), donde PEEK(&HF3B0) nos devuelve el ancho establecido por WIDTH. Por ejemplo: si trabajamos con un ancho de 37 (el estándar cuando encendemos el equipo), quedan 2 columnas a la izquierda - no disponibles - y 1 columna a la derecha.

ENFILAR Y ENCOLUMNAR

Para poder visualizar más fácilmente los efectos de la función GCHAR, preparamos esta sencilla subrutina que denominamos "ENFILAR Y ENCOLUMNAR", cuya misión consiste en colocar en la columna 0 (no disponible para el usuario; trabaja con un "WIDTH 37") el segundo dígito de la FILA correspondiente; y en la fila 23, cuya dirección comienza en 920 (ver Figura 4), también el segundo dígito pero de la COLUMNAS.
Nótese cómo, con un poco de imaginación, podemos fácilmente expandir notablemente los recursos -escasos- que nos brinda el BASIC.

EL LLENADO DE LA PANTALLA

Al efecto de facilitar aún más la comprensión del uso y las posibilidades que brinda la función GCHAR, agregamos esta subrutina auxiliar de la figura 5, que llena varias filas (16) de la pantalla con letras al azar, lo cual sumado al encolumnamiento y al enfilamiento, nos permitirá ejercitarnos adecuadamente.
Obsérvese que en la línea 70 los límites de la pantalla disponible, cuando trabajamos con un ancho de 37 columnas (WIDTH 37), están dados por las columnas 2 (la tercera empezando desde el margen izquierdo) y por la 38

FIGURA 6

```
82 REM SUBR SALIDA IMPRESA
84 REM --------------------
86 FOR F=FI TO FF
88 FOR C=CI TO CF
90 LPRINT CHR$(FN GCHAR(C,F));
92 NEXT C
94 LPRINT CHR$(10);
96 NEXT F
98 RETURN
```

(la segunda desde el margen derecho).

LA SALIDA POR IMPRESORA

Esta subrutina cumple en brindarnos una copia impresa del texto existente en la "VENTANA" delimitada por el usuario. Estrictamente hablando no es este el concepto adecuado de ventana, sino que más bien deberíamos referirnos a "PANELES".
Para quienes aún no sean propietarios de una impresora, sugerimos no amedrentarse por ello y concurrir, munidos de un paquete de masas para la secretaria, al CENTRO DE ASISTENCIA AL USUARIO de TALENT MSX más cercano a su domicilio, llevar el programa grabado en casete o disquete, y probarlo.

DESAFIOS A LA IMAGINACION

Queda planteado un desafío para los lectores de **LOAD**, que consiste en perfeccionar la Subrutina de SALIDA IMPRESA, de modo tal que respete en el papel, la posición relativa del PANEL delimitado en la pantalla.
Otra propuesta es la de reprogramar alguna tecla de función para que, al pulsarla, se cargue (MERGE) una rutina como la que hemos presentado, obviamente sin la parte de llenado de pantalla, con lo cual se podría hacer un vuelco por impresora de todo o de parte de la pantalla.
En este caso se deberían reubicar los mensajes de las líneas 30 a 36, para que no interfieran con la pantalla.
Otra posibilidad sería la de reprogramar otra tecla de función para que nos permita almacenar en casete o disquete la pantalla de nuestro interés, siempre recurriendo a la función GCHAR que hemos presentado.
Esperamos recibir propuestas al respecto del Club de Fans de la MSX. Las iremos comentando en futuras colaboraciones.

(1) - Ver en LOAD MSX de Junio/1987 la Sección BUZON.

Gustavo O. Delfino

CABBAGE PATCH KIDS

Por un largo sendero una niña debe sortear obstáculos para continuar avanzando.
Al comienzo podemos elegir a nuestro personaje (cara, peinado, vestido y nombre). Todo al compás de una de las pegadizas melodías que tiene el programa.

Anna Lee, que es el nombre de la niña si no lo modificamos, debe ir de pantalla en pantalla evitando ser arrollada por las piedras que vienen rodando y saltando por sobre un perro que está tendido en el medio del camino.

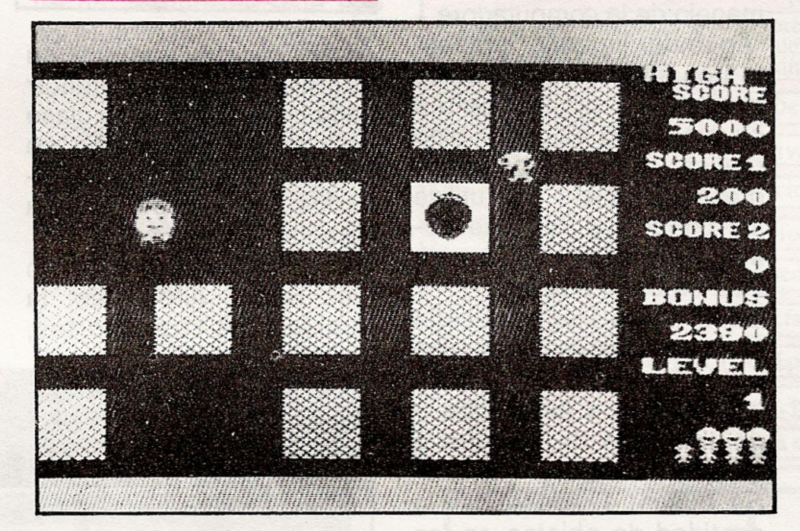
Hay también otro tipo de obstáculos que están situados en serie: troncos y pozos. Además hay que pasar un lago viajando en una isla flotante, a la que accedemos mediante una liana.

Otras pantallas tienen obstáculos combinados. Se llega a la última del primer nivel (9), que consiste en cuatro trampolines sobre los que hay que saltar para recoger una fruta.

Cumplido esto accedemos al siguiente nivel no sin antes hacer unos saltitos de festejo. Las complicaciones, como es de suponer, van en aumento. Aparece un pájaro con cara de malo, una fogata y fuentes que debemos saltar. También hay visitantes nuevos como unos peces saltarines que debemos evitar en la pantalla de pozos, o unas arañas que quieren posarse sobre nuestra cabeza.

Párrafo aparte merece la música. Pegadiza, melodías variadas y efectos sonoros oportunos hacen intuir un trabajo muy serio en la realización de este programa. Es un juego muy "piola" para chicos de 10 a 12 años, pero se pueden "prender" los mayores si se resignan el pasar por las primeras pantallas que son muy fáciles. (MICROBYTE)

PAIRS



Un juego del tipo del "encuentre su pareja", en el que debemos ir dando vuelta los diferentes bloques (de a uno) y memorizando qué figuras se hallan debajo de ellos, para que, al descubrir dos iguales, y destaparlos en forma consecutiva, los mismos sean retirados del tablero y se nos otorguen puntos por ello.

Para limpiar todo el panel hay un tiempo límite. El excedente de ese tiempo también se vuelca en nuestro puntaje en forma de "bonus", por lo que debemos tratar de apurarnos.

Otra dificultad es que hay un monstruito (a la manera de los Pac-man) que nos va cerrando el camino con la "non sancta" intención de comernos (devorarnos, deglutirnos, en otras palabras, que seamos su banquete).

Los diferentes niveles del juego alternan tres clases de dibujos (frutas, hortalizas y animales) lo que permite a los chicos ir familiarizándose con estos conjuntos a la vez que ejercitan la memoria visual.

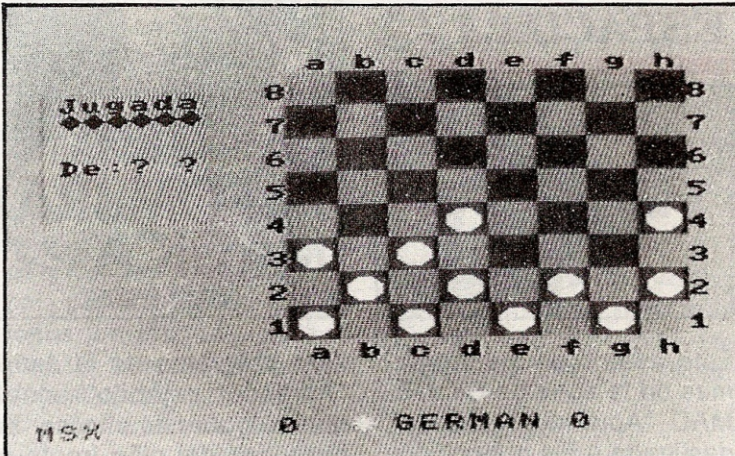
Los bloques a destapar son veinte: nueve parejas de objetos, un comodín, y un objeto que queda circunstancialmente libre y hace pareja con el comodín.

A medida que superamos niveles, los hambrientos monstruos se hacen más inteligentes (nos acorralan mejor), más veloces y, lo que es peor para nosotros, crecen en número.

Pueden jugar una o dos personas (al mismo tiempo), siendo esta última variante muy entretenida ya que fomenta un sano espíritu de competencia.

Sobre la música podemos decir que comenzamos odiándola pero cuando terminamos de jugar ya estábamos tarareándola. (MICROBYTE)

DAMAS



Sentarse frente a la computadora y tratar de derrotarla es la propuesta de este programa.

El juego es las damas, más precisamente la versión del mismo que se juega sobre un tablero de ocho por ocho (como el del ajedrez).

Al comenzar nos pregunta nuestro nombre y si queremos jugar primero. Luego el juego se desarrolla normalmente, siendo bastante interesante el nivel de juego de la máquina, lo que le posibilita dar varios sustos, y hasta gana, si uno se descuida.

El programa tiene además una protección para que no podamos ingresar jugadas imposibles. Las reglas del juego son las tradicionales. Si no comemos una ficha enemiga que está en el aire, la máquina automáticamente "soplará" por no haberlo hecho y nos quitará nuestra ficha (¡mala!).

Cuando llegamos con una ficha propia a la octava fila, ésta se cambiará por una dama.

Pudo haberse mejorado un poco la presentación del juego colocando instruccio-

nes y diagramando un poco mejor la pantalla de presentación. Pero el programa cumple su objetivo, que es presentarnos lucha en una partida de damas, por lo que lo recomendamos a los que gustan de este juego. (MICROBYTE)

LA MOTO ESPACIAL 1



Este es uno de los programas de la serie de SYSTEMAC "Aprender jugando", destinada a los más chiquititos. Esta serie consiste en un conjunto de programas educativos dirigidos a niños de cuatro años en adelante. Asimismo algunos juegos pueden ser utilizados por niños de tres años, como también por aquellos que, superando los seis años presenten problemas de aprendizaje... La intención de esta serie es que los niños desarrollen su lenguaje y adquieran conocimientos al mismo tiempo que juegan con su computadora.

La Moto Espacial 1 pretende desarrollar la capacidad de observación en base a la formación de conjuntos.

Al comenzar el juego se propone una actividad. Luego, una vez resuelta, el niño recibe informaciones visuales y sonoras que le indican aciertos y errores. En el caso de que se equivoque, el niño tendrá otra oportunidad (o varias si es necesario), hasta que logre un acierto.

Así irá descubriendo que un conjunto es una colección de objetos que tienen una característica común. También desarrollará su destreza manual y su sentido espacial al tener que mover una nave para posarla sobre el objeto elegido y transportarlo por una pantalla hasta el lugar adecuado. (SYSTEMAC)

LA ALFOMBRA MAGICA 1



Otro programa de la serie "Aprender jugando" en el que se pretende eliminar la pasividad del niño frente a la máquina y estimular la participación activa en el juego. Simultáneamente se le enseña y se lo adapta al manejo de la computadora.

"Aprender jugando" presenta una serie de actividades graduadas por dificultad cuyo objetivo es que el niño realice un aprendizaje progresivo, es decir, que utilice los conocimientos que posee como base del aprendizaje de otros nuevos.

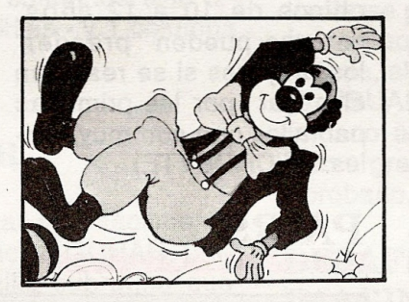
La Alfombra Mágica 1 tiene como objetivo iniciar al niño en la expresión numérica por medio de juegos de identificación de las nueve primeras cifras.

Se busca: 1. Identificar la cantidad de objetos en los conjuntos dados. 2. Reconocer las cantidades que representan las grafías de los números. 3. Asociar conjuntos de objetos al número que lo representa. 4. Desarrollar la coordinación óculo-motriz.

El niño recibe información de sus aciertos o errores por medio de un código de color: verde si no comete errores; amarillo, si sólo comete uno, y rojo, si se equivoca más veces. Este código sirve además, como es evidente, para que el niño vaya conociendo el código de señales de tránsito. (SYSTEMAC)

AVENTURAS EN EL CIRCO 1

También de la serie "Aprender jugando". Aventuras en el Circo 1 pretende desarrollar la capacidad de observación por medio de juegos de diferenciación de colores. El objetivo de este juego es que el niño forme parejas de objetos iguales en todas sus cualidades (color, forma, tamaño , posición). Al emparejar objetos se prepara para realizar actividades de clasificación posteriormente. Contribuye a ello, además, el dibujo



simbólico que aparece en pantalla para representar cada tema (conjunto o clase de objetos) por medio de distintos elementos relacionados entre sí por alguna característica común.

Los objetos se agrupan en diez temas diferentes: la familia, los transportes, prendas de vestir, alimentos, flores, animales salvajes, animales domésticos, la casa, la escuela y la ciudad.

El payaso Plín Plín deberá trasladar los objetos de un lado a otro tratando de esquivar pulgas, cosa que hace saltando sobre ellas.

El juego tiene varios niveles, cada uno más difícil que el anterior. (SYSTEMAC)

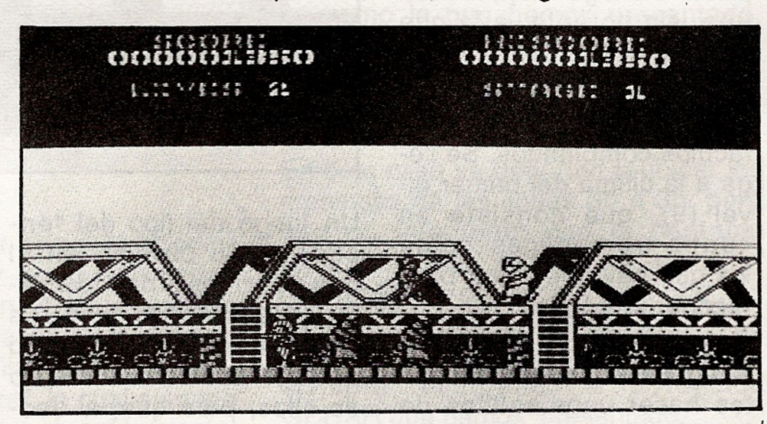
GREEN BERET

Este juego de combate consiste en rescatar a soldados que han sido capturados por el maldito enemigo. Para cumplir esa misión debemos enfrentarnos con los agentes y las tropas que tienen órdenes de destruirnos.

Los soldados armados nos lanzan granadas y nos disparan a quemarropa. A ellos hay que esquivarlos, agachándose cuando vienen las balas y golpeándolos letalmente cuando los tenemos cerca. Luego están los agentes secretos. Unos, los oscuros, quieren prendernos por las buenas. A éstos es bastante fácil eliminarlos con uno o dos golpes. Pero luego aparecen los otros, blancos, que cuando nos ven saltan sobre nosotros cual tigres. La única forma de eliminarlos a estos agentes es evitar que nos ganen de mano en el salto.

El juego presenta mucha acción, en variados escenarios. Cada pantalla tiene un piso superior y uno inferior, y unas escaleritas que nos permiten pasar de uno a otro.

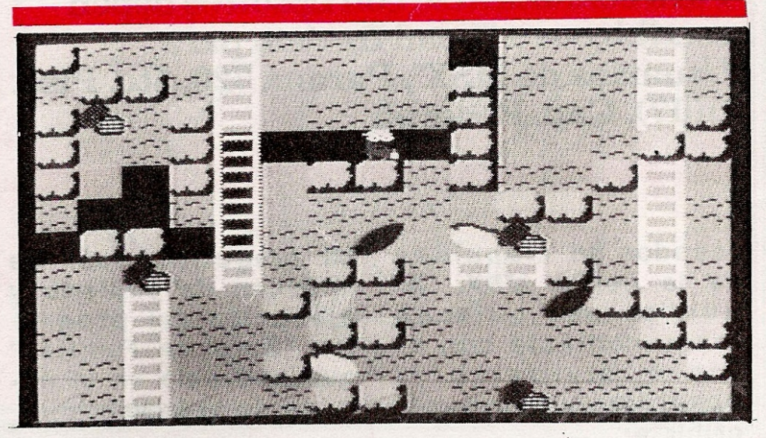
La estrategia adecuada es



combinar el tránsito hacia el objetivo alternando los dos pisos, y luchar sólo cuando sea necesario.

Los juegos de este tipo tienen una gran aceptación y éste, creemos, tiene las condiciones para "capturar" adeptos. (MICROBYTE)

MOLE MOLE 2



Un gordito narigón debe ir recogiendo los más variados objetos, la mayoría comestibles: bananas, cerezas, pedazos de tartas, diamantes, entre otros.
Su camino lo debe establecer atravesando fardos de pastos, que va deglutiendo. El juego es semejante al Boulderdash, pero aquí el camino del hombrecito no es tan libre. Por ejemplo, si quiere subir sólo puede hacerlo por las escaleras, por lo que el peligro de quedar atrapado en la parte inferior de la pantalla es muy grande.
Hay que pensar bien el recorrido que vamos a efectuar. Para eso tenemos un mapita al que accedemos apretando el botón del joystick.
Los niveles del juego aumentan en dificultad a medida que avanzamos, pero la buena noticia es que podemos elegir por cuál nivel empezar (no debemos "tragarnos" las mismas pantallas siempre).
(MICROBYTE)

ADMINISTRADOR DE VIDEO-CLUB

La proliferación del video en los últimos años es comparable en su campo al que tuvo la computación en el suyo.

No es raro entonces que muchos video clubes hayan decidido instalar ya una computadora que maneje la información de que disponen. Sin embargo en muchos casos la instalación de un sistema de este tipo es muy "casera" y los problemas comienzan a aparecer.

LOGICIEL ha creado este sistema administrador de video club que está diseñado para trabajar con un catálogo de hasta 1500 títulos de películas y un archivo de hasta 500 clientes.

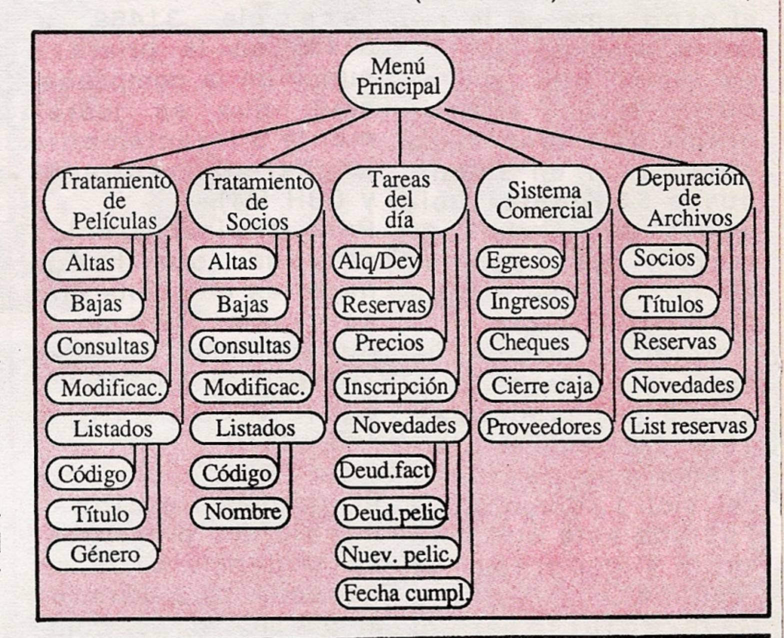
Dispone de todas las posibilidades necesarias para el buen manejo de cualquier video club: altas, bajas, consultas y modificaciones de películas y clientes; posibilidad de incorporación de nuevos clientes con o sin inscripción; listados alfabéticos y numéricos de clientes y títulos; listados de títulos por género; alquiler y devolución de títulos controlados por fecha; control listado de reservas; control de inscripciones; emisión de factura; listados de cumpleaños de los clientes, de deudores de películas y facturas, y de nuevas películas; depuración de archivos; posibilidad de manejo con hasta seis listas estándar de precios más una especial; control de ingresos y egresos, de cheques y de recargos por mora en la devolución; manejo de proveedores; cierre de caja del día; calculadora en todos los menúes; posibilidad de aceptar un módulo de estadísticas.
El sistema consta de dos tipos de pantallas: unas donde se selecciona la aplicación deseada (menúes) y otras donde se ingresan datos para su procesamiento (pantallas de entradas de datos).

En el diagrama vemos al sistema en su conjunto y, de manera clara, la forma en que trabaja.

Restan hacer algunas aclaraciones. En el rubro "reservas" se ingresa desde qué día a qué otro se produce la misma (por ejemplo, desde el 10 hasta el 12). El sistema automáticamente volverá a habilitar la película para su alquiler si el día fijado (10) no fue retirada la misma.

Las seis listas de precios están pensadas para que pueda manejar precios distintos por 24, 48 y 72 horas.

(LOGICIEL)



SINTETIZADOR DE VOZ

Recibimos varias cartas de lectores que han tenido problemas con este programa. Hemos trasladado las dudas al autor del mismo, y aquí está su respuesta: Debido al creciente rumor de las huestes enfurecidas que reclaman mi cabeza he decidido escribir estas líneas para salvar mi integridad física y calmar a los lectores, víctimas de la lluvia de DATA's de dicho programa.
Fernando Ramón y Luis López, de Necochea, son poseedores de una TOSHIBA HX-20, y cuando corren el programa se le resetea la máquina.
Es lógico que esto ocurra, ya que el programa esta preparado para correr en Talent, cuyos 64 K están ubicados en el slot número 1 mientras que en la Toshiba dicha memoria ocupa otro slot (el 3). Al ejecutarse la rutina de la figura 5 (ver programa), este traslada el programa a la página 1 del slot 1. Como en la Toshiba este slot esta vacío el programa se "esfuma".
Para solucionar el problema deberán hacer lo siguiente: reemplazar en la línea 600 del listado BASIC, contando desde la posición del primer DATA (03), el valor 54 por FC (posición 22), 50 por F0 (posición 28), y DA por FA (posición 34). Estos cambios habilitan el slot número 3.
Yendo a la inquietud de Leornado di Biase, de Capital, seguramente en su caso esta faltando algún DATA, lo que hace que la suma de control adicione algún valor incorrecto, produciendo una suma fuera de rango (OVERFLOW).
Con respecto a la misiva de Roberto Chustek, de Tucumán, para su pesar le informo que el programa ha sido correctamente transcripto en la revista, y que el mismo, aunque parezca mentira,...FUNCIONA!!
Personalmente me he tomado el trabajo de controlar toda las DATA's, no encontrando ninguna diferencia con el listado original. Al efectuar dicha comprobacion noté que puede confundirse fácilmente el número "8" con la letra "B". Tomen precauciones con este detalle. Casualmente Roberto tiene una diferencia 31469 a 31466 en la primera suma de la comprobación, que es justamente 3 (exactamente la diferencia entre 08 y 0BH (11).
Para terminar la única salvedad que he de hacer es que falta una sentencia STOP en la línea 80 (80 IF S<>T THEN BEEP:PRINT"* ERROR BLOQUE *:"M:STOP).
Este "BUG" no afecta el normal funcionamiento del programa. La sentencia STOP se necesita para detener al BASIC si se comprueba alguna diferencia en las sumas de verificación.
Espero haber disipado las dudas de muchos y que todos puedan hacer hablar a la "MSX".

EDUARDO LUIS BLOTTA

CORREO INTERNACIONAL

Recibimos desde España una carta de un grupo de jóvenes interesados en máquinas de la norma MSX. Aquellos que quieran tener contacto con ellos e intercambiar programas deben escribir a:

GABRIEL MEDINA VILCHEZ
c/PERU ,6, 1º
18600 MOTRIL
(GRANADA) ESPAÑA

Para comunicarse con nosotros deben escribirnos a 'Load Revista para Usuarios de MSX", Paraná 720, 5to. Piso, (1017), Cap. Federal.

CONCURSO

Me gusta mucho su revista y sobretodo los concursos. Mi pregunta es sobre el tercer concurso de programas. Me gustaría saber que factores se tienen en cuenta para determinar a los ganadores.

PABLO C. SBARRA
Tandil

Load MSX

Los ganadores se determinan luego de un análisis integral del programa. Previamente se hace una selección en base a calificacion de distintos ítems como técnica de programación, documentación, utilidad, originalidad, efectos gráficos y sonoros entre otros. Para mayor información te recomendamos que leas las bases del concurso en donde se da una serie de requisitos que deben cumplir los programas.

BALANCE FAMILIAR

Soy lector de la revista desde hace algunos números y los felicito por el nivel que tiene. Mi pregunta ¿cuál es el orden de copiado y ejecución del programa "Balance Familar" que apareció en el número de agosto?

ALEJANDRO FARIAS
SALTO (BS AS)

Load MSX

El programa que mencionas está compuesto por tres bloques. El orden de copia no interesa pero recuerda hacer una copia en el disco de cada uno de ellos con el nombre que figura en el encabezamiento de cada listado: "balance.ini", "balance.ing" y "balance. cal".
Luego se ejecuta "balance.ini" y por el menú de opciones del mismo se accede a los otros dos bloques. Una vez ejecutado "Balance.ing" se vuelve automáticamente al "balance.ini", y si se quiere que suceda lo mismo con "balance.cal" hay que agregar en este programa la línea 375 RUN"balance.ini"

CONSORCIOS

Luego de copiar el listado del programa "Consorcios" y tratar de ejecutarlo encontré que el mismo se detiene por un error en la línea 2630. ¿Me podrían indicar cuál es?

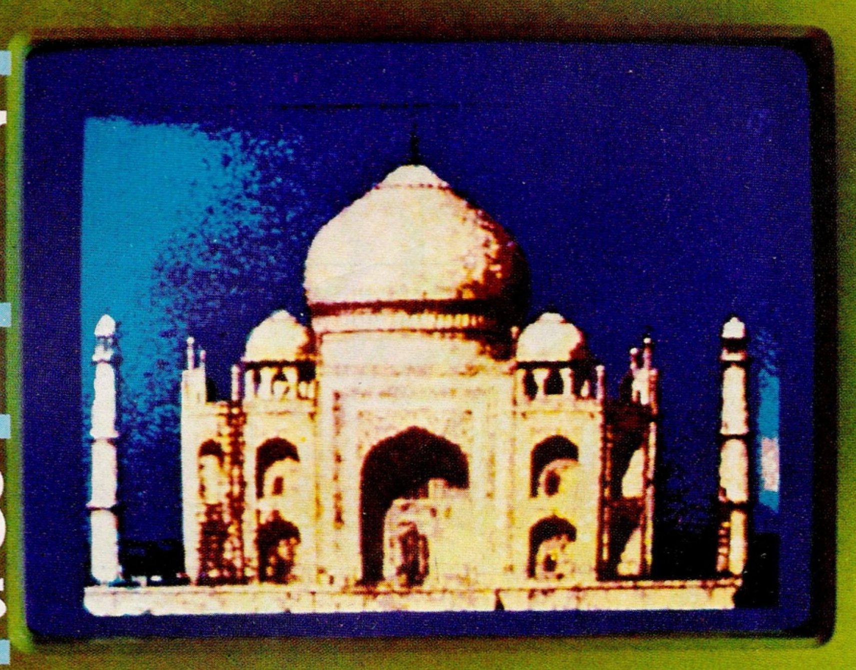
VERONICA GAMBINA
Jose Ingenieros
(Bs As)

Efectivamente se deslizó en el listado un error en la línea que mencionas. Debe ir GOTO 2650 en vez de GOTO #.
Aprovechamos para comentar que la línea 2740 debe ser 2740 GOTO 290



OCTUBRE 1987

IMAGEN DIGITALIZADA



NUEVAS

MSX-2 y D-C128 ARGENTINAS

SORTEO Y CONCURSO

D-C 64/128

HACKERS

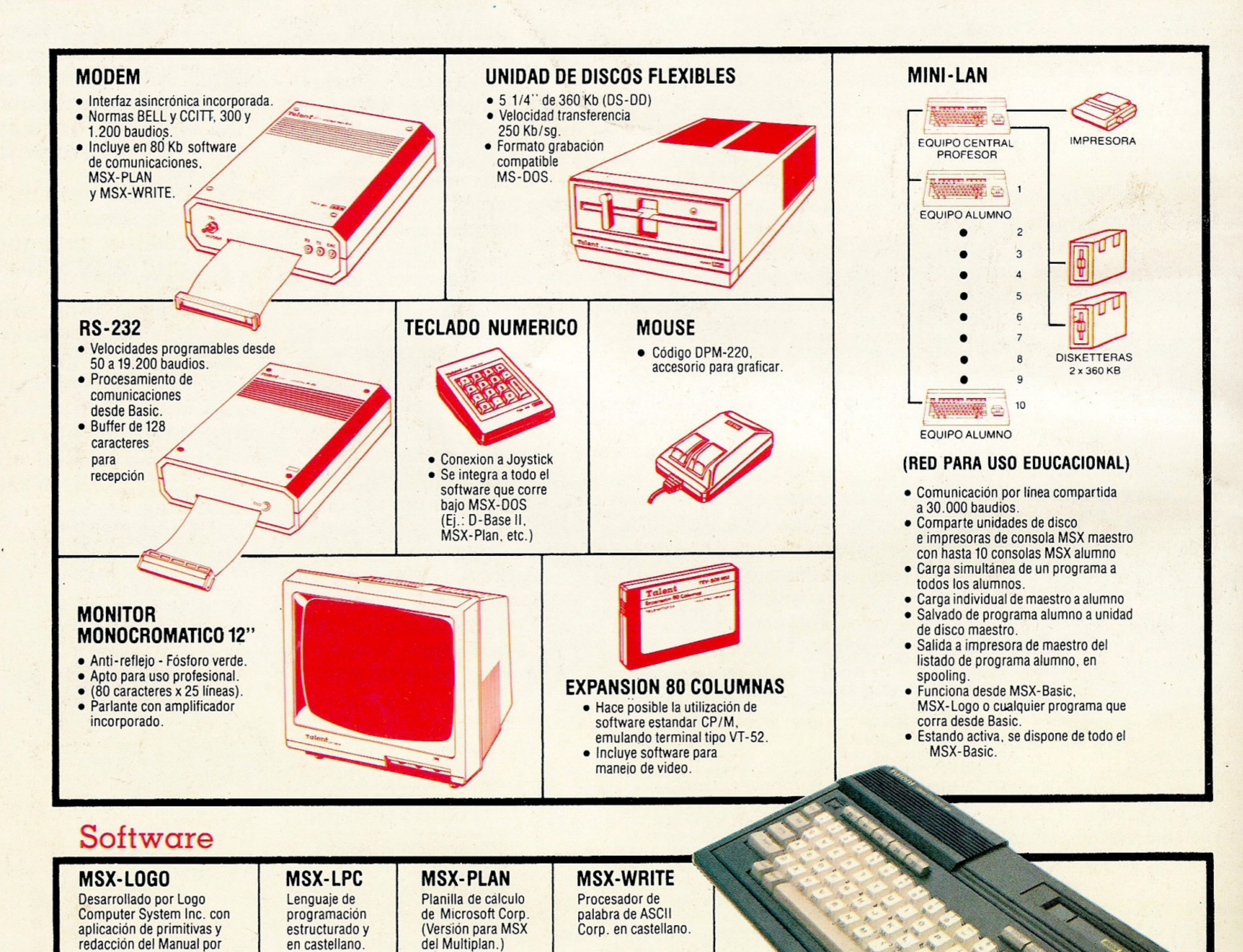
SPECTRUM-TK90

COMUNICACIONES

HARD y SOFT PARA ATARI, SPECTRUM, CZ, TK, DREAN COMMODORE, TI y MSX

AÑO 3 N° 31 ₳ 6.- REP. ARGENTINA

Encienda una computadora Talent MSX y sus periféricos.



Talent

Tecnología y Talento *en su casa*

Producida en San Luis por Telemática S.A. licenciataria exclusiva de Microsoft Corp. y ASCII Corp. para uso de la norma MSX en Argentina.
6 meses de garantía y mensualmente en su quiosco la revista Load MSX.